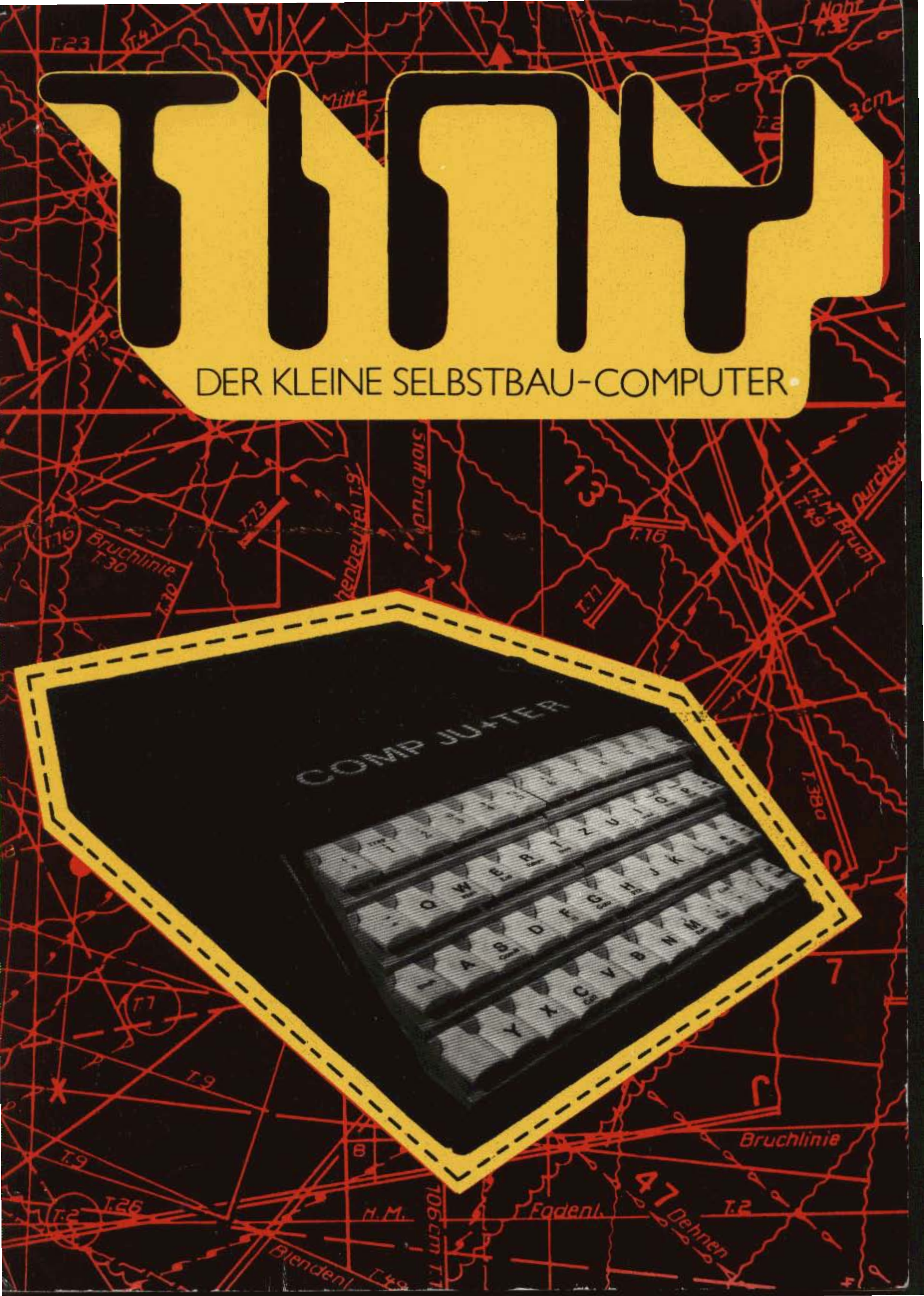

TINY
DER KLEINE SELBSTBAU-COMPUTER
COMP JU+TER

Helmut Hoyer/Norbert Klotz

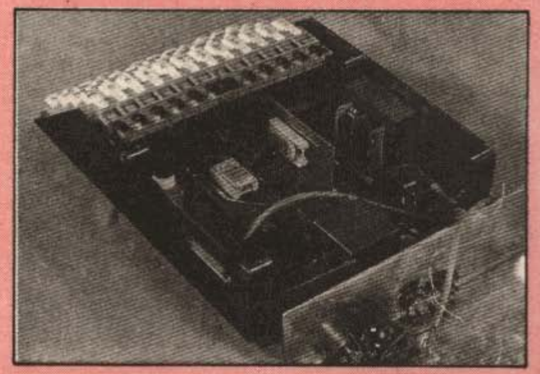
TINY

der kleine Selbstbau-Computer

Vom TINY-BASIC bis zum Entwicklungssystem: Computerbauanleitung für den Einstieg in die Mikrorechentechnik

Verlag Junge Welt GmbH, Berlin

Inhaltsverzeichnis





MASTER MIND
4 MAL 1 AUS 6
1:2222 *
2:2333 O
3:4244 **
4:4255 ***
5:

Klein, einfach, preiswert, modular und nachbausicher präsentiert sich der selbstgebaute Computer.

Für die Bildausgabe eignet sich ein Fernsehgerät.

Spiele in TINY-MP-BASIC, verbunden mit akustischer Ausgabe, erleichtern den Einstieg.

```
PROG 01000000
E00A 32 20
E00C A0 E0
E00E EB :ZKL
WRT: 3A :WRT
E012 4A :WRT
E014 8B :ANF
E00C
```

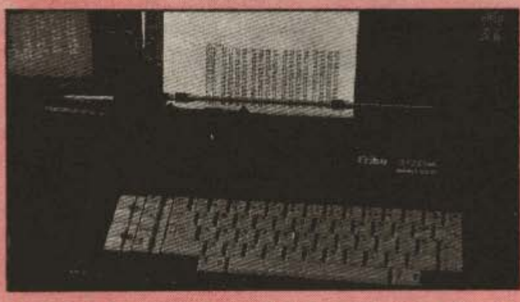
```
EMR-ES 1988

B BASIC
I INIT
S SAVE
L LOAD
R DATA
P PROG
```

Programmieren in Maschinensprache nutzt die Fähigkeiten des Prozessors am besten.

Das Anfangsmenü: Sechs verschiedene Dienste bietet das Betriebssystem.





Grafische Ausgaben sind ohne zusätzliche Hardware möglich.

Druckeranschluß und EPROM-Programmierzusatz werten das Gerät auf zum Entwicklungssystem.

Liebe Leser!

Mit diesem Büchlein stellen wir Euch einen Kleincomputer vor, den Ihr mit sehr geringem Aufwand nachbauen könnt. Und er hat weitere Vorteile: er ist preisgünstig, leistungsstark, ausbaufähig und nachbausicher. Er macht Euch mit der Mikrorechner-Gerätetechnik bekannt und erleichtert Euch den Einstieg ins Programmieren. Fürs erste kann man mit TINY-MP-BASIC anfangen. Getestete Programme liefern wir Euch gleich mit. Später läßt sich der Computer auch ausbauen bis zum Entwicklungssystem.
Jeder, der schon mal selbst beispielsweise einen Rundfunkempfänger gebaut hat, kann sich ohne Bedenken an den Selbstbau des Computers wagen. Wer noch nicht so sicher ist, sollte von unserem Angebot auf Seite 8 Gebrauch machen und die Leiterplatten kaufen, die dann nur noch zu bohren, zu bestücken und in Betrieb zu nehmen sind. Die Bauelemente bietet in der Regel der Amateur-Fachhandel an.
Was kann unser Computer? Eine ganze Menge. Es reicht von Spielprogrammen über das Bekanntmachen mit der Anfängersprache TINY-MP-BASIC bis zum Programmieren in Maschinensprache und zum Einsatz als Steuerrechner zum Beispiel für die Modelleisenbahn. Es können sogar Programme für andere Rechner, die einen Einchipmikrorechner als Basis haben, entwickelt und in EPROM geladen werden.
Bevor Ihr also loslegt, lest dieses Büchlein gründlich. Und Spaß macht das Bauen und Ausprobieren vor allem auch dann, wenn man Freunde findet, die dabei helfen, raten, die Programme tauschen. Gleichgesinnte findet Ihr sicher in entsprechenden Arbeitsgemeinschaften, aber auch in der Zeitschrift JUGEND+TECHNIK, die seit 1987 in ihren Heften zum Selbstbaucomputer Hard- und Softwareangebote veröffentlicht.
Viel Spaß bei Eurem Hobby – und mit unserem kleinen Computer Marke Eigenbau!

Helmut Hoyer und Norbert Klotz

TEIL 1: AUFBAU DES GRUNDGERÄTES

1. Konzeption

Was soll der Computer können?

Das Konzept geht von einem einfachen Grundgerät, aber schon einem Kleincomputer aus, der sich per Zusatzbaugruppen weiter ausbauen läßt. Dabei soll den handelsüblichen Kleincomputern KC 85 oder KC 87 keine Konkurrenz gemacht werden, sondern es steht im Vordergrund, mit möglichst wenig Hardware (Gerätetechnik) auszukommmen. Es schien auch nicht sinnvoll, zu den schon vorhandenen Lösungen eine weitere mit dem Mikroprozessor[1] U 880 D dazuzustellen. Denn der Selbstbaucomputer sollte ja klein, einfach, preiswert und nachbausicher aber dennoch leistungsstark werden.

Am besten kommen Einchip-Mikrorechner[2] diesem Konzept entgegen. Schon mit Bauelementen im Wert von nur etwa 300 Mark, die wir für den Anfang benötigen, läßt sich der Computer aufbauen. Das ist verhältnismäßig wenig im Vergleich zu den Kosten für die benötigten Peripheriegeräte Fernsehempfänger und Kassettenrecorder.



Ansicht des Grundgerätes mit Fernseher

Das modulare Konzept macht recht einfach gestaltete Leiterplatten möglich. Allerdings sind mit dem minimierten Hardwareaufwand auch Kompromisse verbunden. Zum Beispiel macht die relativ geringe Zeichenzahl auf dem Bildschirm Textverarbeitung und Datenbankverwaltung praktisch unmöglich. Dennoch erschließt das Gerät uns ein breites Feld für das Computer-Hobby. Wie andere Kleincomputer kann er Spielprogramme abarbeiten oder Prozesse steuern, zum Beispiel die Aquariumsregelung übernehmen. Nach den ersten Schritten mit der Anfängersprache TINY-MP-BASIC[3] ist auch das Programmieren in Maschinensprache[4] möglich. Schließlich sieht das Konzept vor, solche Zusatzbaugruppen vorzustellen, die das Grundgerät zum Entwicklungssystem aufwerten, mit dem wir Programme für andere Einchip-Mikrorechner entwickeln können. Dazu gehören der EPROM[5]-Programmierzusatz und der Druckeranschluß (vgl. Kapitel 9. und 10.).

[1] **Mikroprozessor** Hochintegrierter Schaltkreis, der die **CPU** eines **Mikrorechners** enthält

[2] **Mikrorechner** Auf der Basis hochintegrierter Schaltkreise (**Mikroprozessor, Speicher,** Peripherieanschlüsse) aufgebauter Digitalrechner mit Ein-/Ausgabegeräten wie Bildschirm, Tastatur, Disketten- oder Magnetbandspeicher, Drucker ...

[3] **BASIC** Einfache bedienernahe Programmiersprache zum Formulieren von Programmen mit kurzen, aus dem Englischen stammenden Wörtern

[4] **Maschinensprache** Formulierungsvorschrift für Programme auf dem Niveau, das direkt von der **CPU** ausgeführt werden kann, meist auf der Basis der **hexadezimalen** Zahlendarstellung.

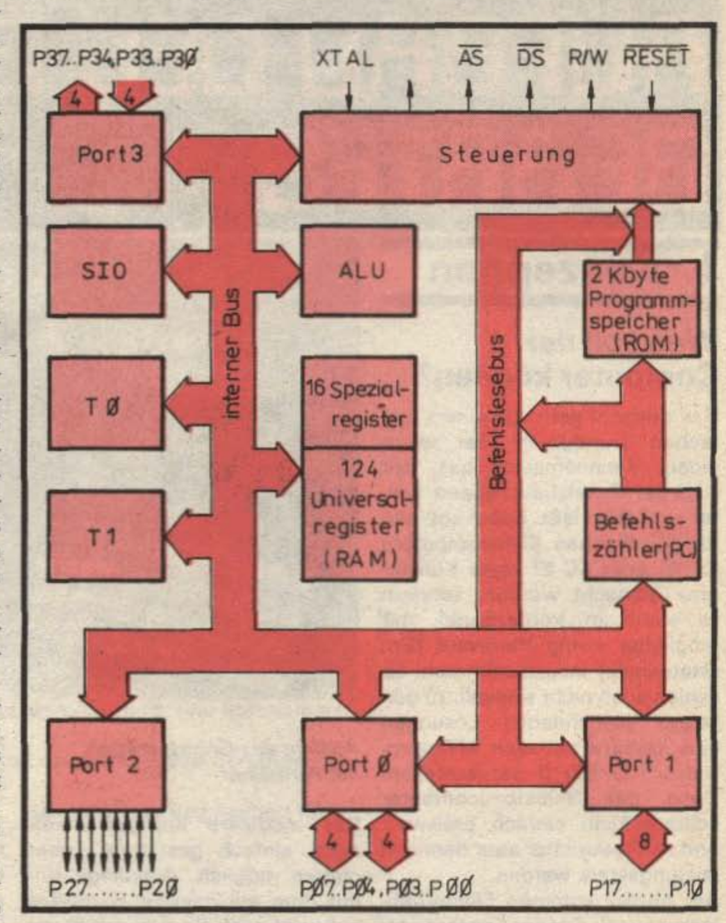
[5] **EPROM** Schaltkreis, der mehrere tausend Speicherzellen enthält, die ihre Information auch ohne Betriebsspannung bewahren und der **CPU** nur Lesezugriffe gestatten. Der Dateninhalt kann mit UV-Licht gelöscht und mit einem speziellen Programmiergerät neu festgelegt werden.

Welcher Prozessor leistet das alles?

Den Kern des Selbstbau-Computers bildet der Einchip-Mikrorechner UB 8830 D (oder UB 8831 D). Er enthält eine 8-bit-CPU[6] (zentrale Verarbeitungseinheit), 2 Kbyte Programmspeicher (BASIC-Interpreter), 140 8-bit-Register[7], vier parallele Ports[8] (Port PØ bis Port P3), zwei Zeitgeber[9] (TØ, T1) und eine serielle Ein-/Ausgabe-Baugruppe (SIO)[10]. **Abb. 1** stellt die gesamte Struktur des Einchip-Mikrorechners dar, die Mikroprozessor, Speicher[11] und Peripherie[12] in einem Schaltkreis vereinigt. Das kommt unserem Konzept entgegen, zumal auch noch der BASIC-Interpreter[13] fest implementiert ist. Detaillierte Beschreibungen zu Einchip-Mikrorechnern geben übrigens das Buch „Einchipmikrorechner" von H. Kieser und M. Bankel, das im VEB Verlag Technik erschien, und das „ABC Einchip-Mikrorechner" in der Zeitschrift „JUGEND+TECHNIK", Heft 7/88 bis 6/89.

Nach außen wirkt dieser Prozessor über 32 Datenpins[14], die den jeweils acht Bitpositionen (2^0 bis 2^7) der vier Ports zugeordnet sind, und sechs Steuersignale. Hinzu kommen Masse- (00) und Betriebsspannungsanschluß (5P). Für das Verwenden der 32 Datenpins bietet der Schaltkreis viele Möglichkeiten. In unserem Computer bilden die Signale P1Ø bis P17 (Port 1) und PØØ bis PØ7 (Port Ø) den Daten- und Adreß-

1 Innerer Aufbau des Einchip-Mikrorechners



[6] **CPU** Zentrale Verarbeitungseinheit (central processing unit) zur Steuerung des Informationsaustauschs über den **Bus** und zur Ausführung von Maschinenbefehlen einschließlich arithmetischer und logischer Verknüpfungen mit einer Arithmetik-Logik-Einheit (ALU)

[7] **Register** Gruppe von acht **Flipflops** zum Speichern der 8 bit eines Bytes (DØ bis D7), der bei 8-bit-Rechnern üblichen Informationseinheit, in der **CPU**

[8] **Port CPU**-nahe Baugruppe für den Anschluß von **Peripherie** mit 8-bit-paralleler Datenübertragung

[9] **Zeitgeber** Baugruppe zum Ableiten von periodischen Signalen für universelle Nutzung aus dem Systemtakt

[10] **SIO CPU**-nahe Baugruppe für den Anschluß von **Peripherie** mit serieller Datenübertragung

[11] **Speicher** Zum Kern von **Mikrorechnern** gehörende Baugruppe, die in vielen Speicherzellen je 1 byte aufbewahrt. Der Auswahl der jeweils mit dem **Bus** zu verbindenden Speicherzelle dient die von der **CPU** erzeugte Adresse.

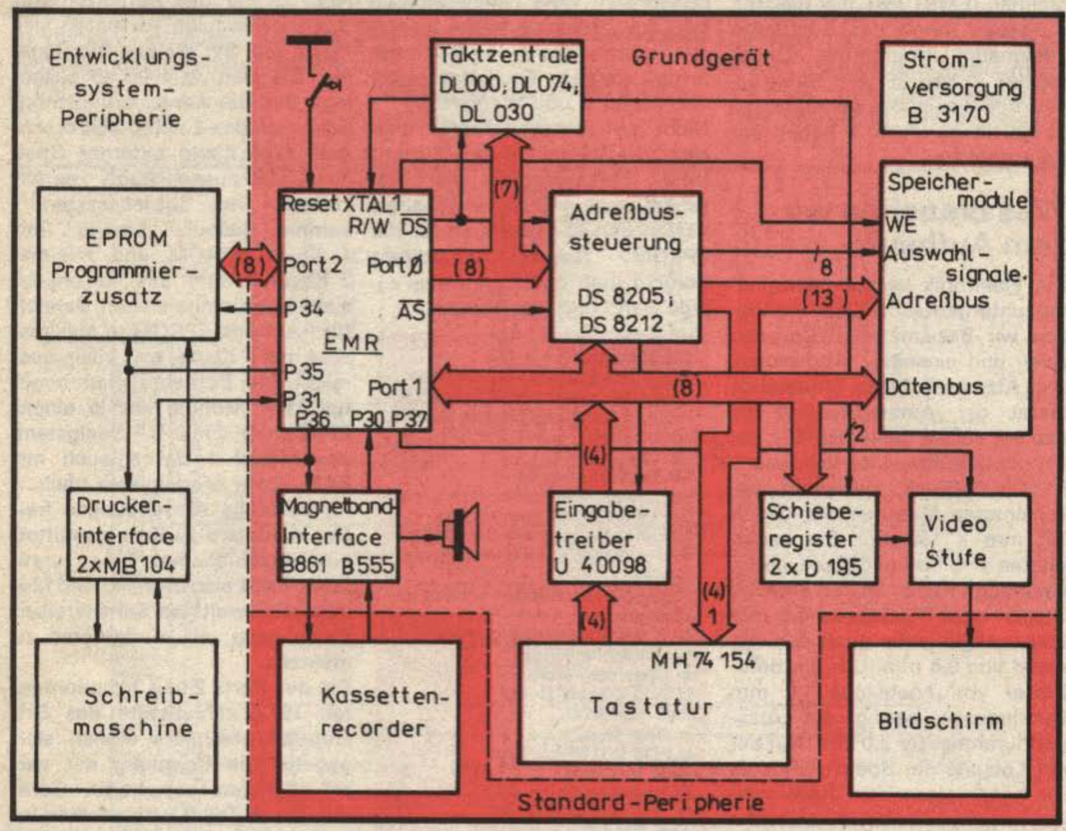
[12] **Peripherie** Gerätetechnik von Rechnern außerhalb der zentralen Baugruppen (**CPU**, **Speicher**, Peripherieanschlüsse), die der Ein- und Ausgabe, als Massenspeicher (Diskette, Magnetband) oder der Datenübertragung zu anderen Rechnern dient.

[13] **Interpreter** Übersetzungsprogramm, das in höheren Sprachen (z. B. **BASIC**) formulierte Anweisungen von der **CPU** ausführen läßt, indem es entsprechende Maschinenprogramme zuordnet. Im Gegensatz zum Compiler, der das Programm vor der Ausführung komplett in **Maschinensprache** übersetzt, behandelt der Interpreter stets nur die gerade auszuführende Anweisung.

[14] **Pin** Anschlußstift eines Schaltkreises. Die Numerierung erfolgt (von oben gesehen) beginnend mit Pin 1 an der der Gehäusemarkierung (Aussparung oder Aufdruck) nächsten Ecke entgegen dem Uhrzeigersinn.

[15] **Takt** Periodische Schwingung, Basis für Zeitsteuerungen. In Mikrorechnern erzeugt ein Quarzgenerator den Systemtakt (hier: XTAL, 8 MHz), aus dem die **CPU** ihre Aktivitäten und eine zusätzliche Taktzentrale die im jeweiligen Rechner außerdem erforderlichen Steuersignale in bestimmter zeitlicher Folge ableiten.

bus, während Port 2 und Port 3 dem Anschluß peripherer Geräte dienen. Über XTAL 1 wird der Takt[15] zugeführt. Die Steuersignale R/W (read/write = lesen/schreiben), $\overline{AS}$ (adress strobe = Adressen gültig) und $\overline{DS}$ (data strobe = Daten gültig) organisieren die Informations-



2 Übersichtsschaltplan des Computers

Abb. 3: Bauelemente für Grundgerät mit Modulvariante A ohne Stützbatterie und ohne UHF-Modulator

Schaltkreise:
1 UB 8830D (UB 8831D)
4 U 224D (U 214D)
2 U 2716C (K 573 РФ2)
1 U 40098 D
1 DL 000 D
1 DL 030 D
1 DL 074 D
2 D 195 D
2 DS 8205 D (UCY 74S405)
1 DS 8212 D (K 589ИР12)
1 MH 74154 (K 155 ИД3)
1 B 555 D
1 B 861 D (B 761 D)

Kondensatoren:
1 22 p
1 33 p
2 100 p
2 2,7 n
6 10 n
1 22 n

Widerstände (Bauform 23207):
1 180 Ω
1 270 Ω
2 330 Ω
1 470 Ω
1 560 Ω
1 680 Ω
1 2,7 k
4 3,3 k
1 6,8 k
1 12 k
4 15 k
2 22 k
5 100 k

Elkos (10 V oder mehr):
2 1 µF TGL 38928
1 2,2 µF TGL 38928
2 10 µF TGL 38928
2 22 µF TGL 38928
1 100 µF TGL 38928
1 2200 µF TGL 38908

Transistoren/Dioden:
2 SC 238 d (SC 237 d)
1 VQA 23 o. ä.
1 1 PM 05 (1 PM 4)
2 SAY 30 o. ä.

sonstige Bauelemente:
1 Netztrafo 8V/0,8 A (8,3 V/1,3 A)
1 Feinsicherung T 0,1 A mit Fassung
1 Telefon-Hörkapsel (kleiner Lautsprecher)
1 Diodenbuchse 3- oder 5-polig
1 HF-Buchse 75 Ω
2 IC-Fassungen 24-polig
8 TT-Tastenpulte
1 TSE-Taster (TSS-Taster)
Blech, Schrauben, Schaltdraht usw.
1 Ätz-Set und Cevausit 2-seitig 135 mm × 155 mm **oder** Leiterplatten per Bestellung (vgl. Seite 8)

Schwingquarz:
8000 kHz
oder Kondensator 82 pF

transporte über den Bus[16].

Abb. 2 zeigt alle Baugruppen als Übersicht. Der Einchip-Mikrorechner (EMR) löst die meisten Aufgaben durch das Ausführen geeigneter Programme. Daher werden in den übrigen Baugruppen nur wenig Bauelemente benötigt. In der **Abb. 3** haben wir sie aufgelistet.

Was brauchen wir zum Aufbau?

Die Elektronik ist auf Leiterplatten untergebracht. Dazu benötigen wir Basismaterial (Cevausit zwei- und einseitig), Abdecklack und Ätzmittel. Diese Materialien bietet der Amateurhandel als Ätz-Set für 42 Mark an. Für die Grundplatte brauchen wir zusätzlich ein Stück zweiseitig beschichteten Materials der Größe 135 mm × 155 mm. Die Leiterplatten sind nur mäßig schwierig: Leiterzüge haben an den engsten Stellen eine Stärke von 0,5 mm und untereinander auch den Abstand von 0,5 mm. Um die Bohrlöcher von höchstens 1,0 mm Durchmesser beträgt der Lötaugendurchmesser 2,0 mm. Nur auf der Lötseite der Speichermodule für höher integrierte RAM werden auch Leiterzüge zwischen benachbarten Lötaugen durchgeführt, so daß es hier noch enger wird.

Zum Zeichnen braucht man eine feine Zeichenfeder, zum Bohren einen Spiralbohrer mit 0,8 bis 1,0 mm Durchmesser und eine passende Ständerbohrmaschine. Für das Bestücken werden ein Lötkolben mit kegelförmiger kleiner Spitze (ca. 25 W), Lötzinn und Kolophonium benötigt. Für die Inbetriebnahme brauchen wir einen Spannungsmeser. Ein Oszillograph stellt eine große Hilfe dar, aber auch mit Kopfhörern lassen sich viele Fehler aufspüren. Das Gehäuse sollte wegen der Schirmwirkung aus Blech gefertigt werden. Auch das erfordert Material und Werkzeug.

Nicht zu vergessen sind eine Handvoll Schrauben und Muttern (M3), Steckverbinder für Anschlußkabel, Sicherungshalter, Tasten und IC-Fassungen für die EPROM. Neben Schaltdraht braucht man dann noch eine ruhige Hand und viel Geduld.

Leiterplatten
Begrenzte Stückzahlen amateurgerechter, ungebohrter Leiterplatten zum Computer können bestellt werden bei:
Gerlich
Markscheideweg 08/417
Neubrandenburg, 2000
Die Leiterplatten werden per Nachnahme zugeschickt, etwas Wartezeit müßt ihr allerdings einplanen.
Folgende Leiterplatten können bestellt werden:
- Prozessorplatine (vgl. Abb. III./IV. Umschlagseite)
- Speicherplatine U 2716 C/U 224 D (vgl. Abb. 12/13/14)
- Speicherplatine U 2716 C/U 6516 D (vgl. Abb. 16/17/18)
- Speicherplatine U 2716 C/U 6264 D (vgl. Abb. 20/21/22)
- Vereinfachte Speicherplatine U 2716 C (vgl. Abb. 24/25)
- Ein-/Ausgabeplatine (vgl. Abb. 32/33)
- EPROMMER (vgl. Abb. 50/51)

Weitere Anfragen betreffs Leiterplatten bitte an obige Adresse richten.

Wie läßt sich der Computer erweitern?

Konstruktiver Kern des Comuters ist die Prozessor-Platine. Sie enthält die Elektronik des Grundgerätes ohne Tastatur, Netztrafo und Speicherschaltkreise (vgl. Abb. 2). Für das Aufsetzen von Speichermodulen existieren vier Plätze mit den nötigen Bussignalen. Die Hardware ist so ausgelegt, daß die Adressierungsmöglichkeiten des Einchip-Mikrorechners (124 Kbyte externer Speicher) voll ausgeschöpft werden können. Das Betriebssystem[17] kommt jedoch bereits mit 4 Kbyte EPROM und minimal 2 Kbyte RAM[18] aus. Es organisiert einen einheitlichen Bereich für RAM und EPROM in maximal acht mal 8 Kbyte, was völlig ausreicht. Das Betriebssystem benötigt zwei Module mit je einem EPROM U 2716 C, wenigstens das Modul 1 davon auch mit RAM. Zwei Modulplätze bleiben für beliebige Anwendungen frei. Der modulare Aufbau gestattet, aus verschiedenen Typen statischer RAM auszuwählen und Module mit veralteten Schaltkreisen durch neue mit moderneren zu ersetzen.

Die den Ports 2 und 3 zugeordneten 16 Anschlußstifte des Einchip-Mikrorechners eignen sich gut für die Kopplung mit verschiedensten peripheren Geräten. Die in Teil II vorgestellten Interfaces (Anschlußsteuerungen) für die Schreibmaschine Erika 3004 und das Magnetbandgerät sowie der EPROM-Programmierzusatz können als Beispiele gelten. Es gibt darüber hinaus viele andere Möglichkeiten.

Erweiterungen der Programmtechnik lassen sich in EPROM auf bislang freien Modulplätzen stationieren oder transident (bei Bedarf in den RAM zu laden, z. B. von Kassette) auslegen. Im

---

[16] **Bus** Verbindungsdrähte zum Informationsaustausch der zentralen Baugruppen eines Rechners untereinander. Die **CPU** steuert alle Busoperationen. 8-bit-Rechner verwenden gewöhnlich acht Leitungen als Datenbus (DØ bis D7) zum eigentlichen Informationstransport, 16 Drähte als Adreßbus (AØ bis A15) zum Festlegen des Partners der CPU beim Informationsaustausch (Peripherieschaltkreis oder Speicherzelle) und einige Steuersignale wie **Takt**, Rücksetzen ($\overline{RESET}$), Lese-Schreib-Umschaltung (R/W) u. a. als Steuerbus. Verwendet die CPU Verbindungsdrähte sowohl zur Adreßausgabe als auch für den Datentransport, definieren die Steuersignale $\overline{AS}$ (Adressen gültig) und $\overline{DS}$ (Daten gültig) die gerade vorliegende Nutzung.

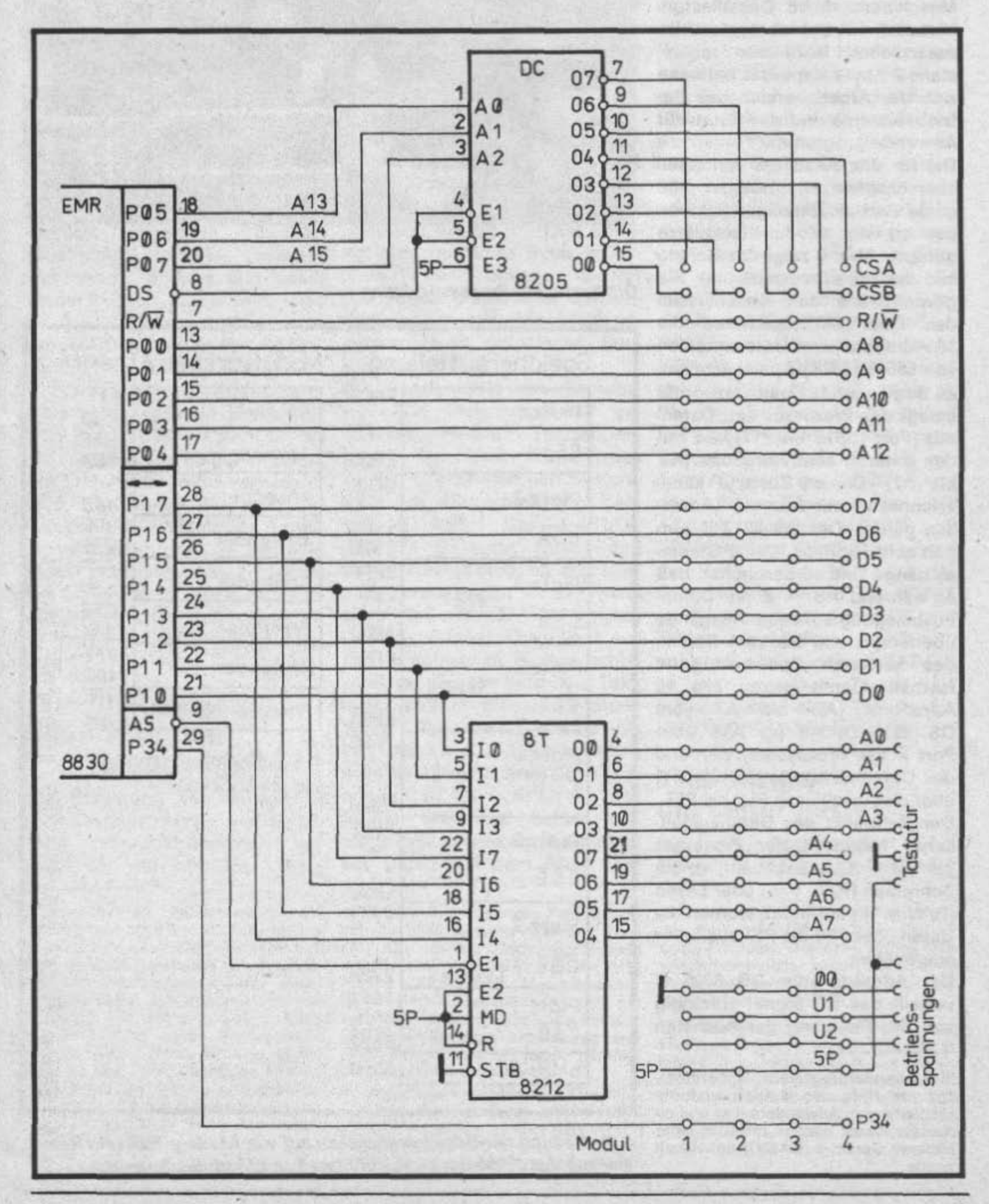
[17] **Betriebssystem** Grundlegende Programme, die für den Betrieb eines Rechners erforderlich sind, wie Bildausgabe, Tastenabfrage, Übersetzer für die Anwendersprache (z. B. **BASIC-Interpreter**).

[18] **RAM** Schaltkreis, der mehrere tausend **Flipflops** (statischer RAM) oder winzige Kondensatoren (dynamischer RAM) zum Speichern je eines Bits enthält. Die **CPU** kann hier sowohl lesen als auch schreiben. Zum Datenerhalt benötigen RAM-Schaltkreise die Betriebsspannung, dynamische RAM zusätzlich Auffrisch-Zyklen (refresh), die die sich sonst selbst löschenden Kondensatoren nachladen.

„Computerklub" der Zeitschrift „JUGEND+TECHNIK" wurden und werden verschiedene Erweiterungen von Hard- und Software zu diesem Computer veröffentlicht (vgl. auch Literaturübersicht S. 64).

2. Prozessorplatine

4 Schaltbild der Adreß- und Datenbussteuerung



Wie wird der Speicher organisiert?

Der Prozessor UB 8830 D (oder UB 8831 D) enthält einen Interpreter (Programm-Ausführung) für die Programmiersprache TINY-MP-BASIC. Um ihn nutzen

zu können, ist die Erweiterung des Speichers mit zusätzlichen Speicherschaltkreisen nötig. Unser Computer benötigt 4 Kbyte EPROM für Ein-/Ausgabe-Programme, den BASIC-Editor (Programm-Eingabe) und die Software für das Programmieren in Maschinensprache. Das alles gehört zum 4-K-Betriebssystem. Im zusätzlichen RAM von mindestens 2 Kbyte Kapazität befinden sich der Arbeitsbereich des Betriebssystems und der Raum für Anwenderprogramme[19].

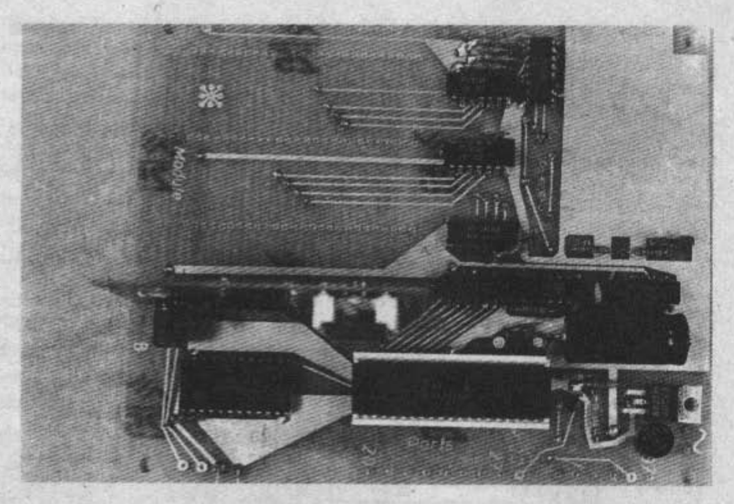
Die für den Anschluß von Speicherschaltkreisen nötigen Signale sind als 29poliger Systembus an vier Modul-Steckplätze geführt. **Abb. 4** zeigt das Schaltbild der Adreßbussteuerung. Sie gewinnt aus den Anschlüssen des Einchip-Mikrorechners die 16 Adreßbits zum Unterscheiden von 65536 (64 K) Speicherzellen. Zu Beginn jedes Speicherzugriffs belegt der Prozessor den Datenbus (Port 1, P1Ø bis P17) kurz mit den unteren acht Adreßbits (AØ bis A7). Diesen Zustand kennzeichnet er mit $\overline{AS}$ = Ø (Adressen gültig). Der DS 8212 D enthält acht Flipflops. Seine Steuereingänge sind so beschaltet, daß er während $\overline{AS}$ = Ø die Datenbusbelegung auf seine Ausgänge übernimmt und bis zum Beginn des nächsten Speicherzugriffs festhält. Damit gelten alle 16 Adreßbits (AØ bis A7 vom DS 8212 D, A8 bis A15 über Port Ø des Prozessors) während des Datentransports (DØ bis D7) über die Anschlüsse P1Ø bis P17.

Den Zeitpunkt des Datenaustausches bestimmt der Prozessor mit $\overline{DS}$ = Ø. Je nachdem, ob ein Schreiben (R/W = Ø) oder Lesen (R/W = 1) stattfindet, werden die Daten über P1Ø bis P17 aus- oder eingegeben.

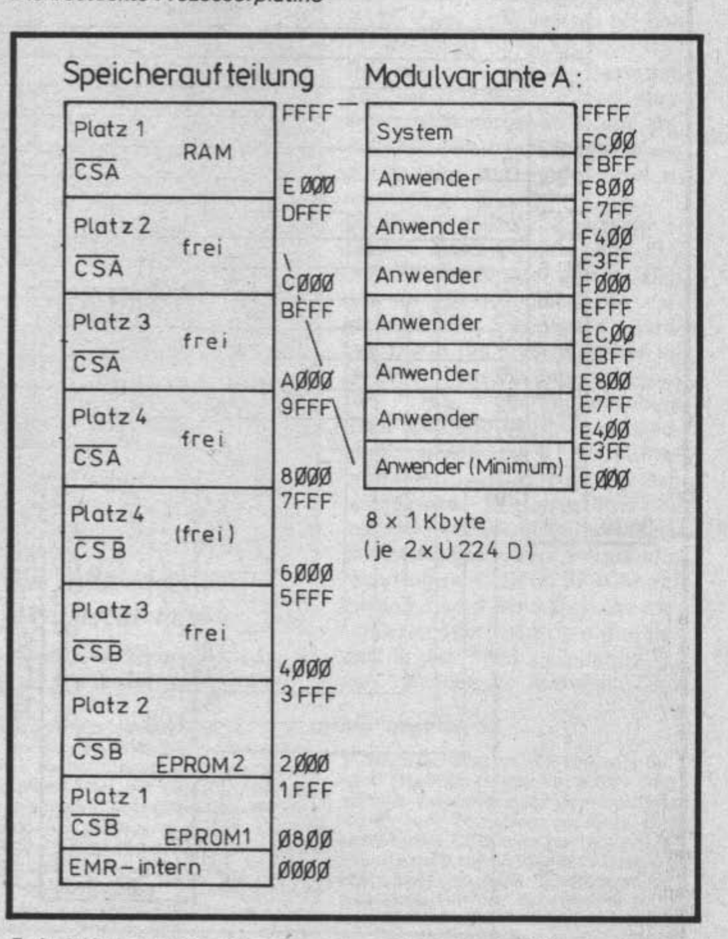
Der Adreßdekoder DS 8205 D verteilt das $\overline{DS}$-Signal abhängig von der Belegung der höchsten

---

[19] **Anwenderprogramm** Programm, das mit Hilfe des **Betriebssystems** Probleme des Anwenders löst und zu diesem Zweck speziell meist in einer höheren Sprache (**BASIC**) entwickelt wurde.



Die bestückte Prozessorplatine



***5** Aufteilung der Speicheradressen auf vier Modulplätze und Realisierung des RAM-Bereichs auf Platz 1 mit Modularvariante A*

drei Adreßbits (A13, A14 und A15) an einen der vier Module. Dabei wird der Speicher für die Module in acht Bereiche aufgeteilt (**Abb. 5**). Die Adressen %ØØØØ bis %Ø7FF (hexadezimal)[20] sind den inneren Baugruppen des Einchip-Mikrorechners vorbehalten und stehen daher dem externen Speicher nicht zur Verfügung.

Jeder Modul erhält zwei Auswahlsignale des DS 8205 D, die direkt als $\overline{CS}$ (chip select: Schaltkreisfreigabe) und $\overline{OE}$ (output enable: Ausgangsaktivierung) der Speicherschaltkreise verwendet werden können. Diese Organisation gestattet Speicherschaltkreise mit einer Kapazität von maximal 8 Kbyte. Auf jedem Modul können davon zwei, also bis zu 16 Kbyte untergebracht werden. Der Bereich von %6ØØØ bis %7FFF von Modul 4 läßt sich jedoch ohne zusätzliche Hardware nicht nutzen, da dieses $\overline{CSB}$ (von Pin 12 des DS 8205 D) für die Freigabe des Eingabetreibers des Tastatur-Interface verwendet wird. Das Signal P34 kann der Prozessor als zusätzliches Adreßbit verwenden, was 32 Kbyte je Modul realisieren läßt. Für unseren Computer ist diese Möglichkeit jedoch ohne Bedeutung, da so viel Kapazität nicht benötigt wird.

Wie funktionieren Bilderzeugung und Tasteneingabe?

Die Steuerung von Bildschirm und Tastatur wurde weitestgehend programmtechnisch gestützt, um mit möglichst wenig Bauelementen auszukommen. Bei diesem Konzept lassen sich 13 Zeichen in jeder der acht Bildschirmzeilen darstellen. Nur das Serialisieren der Bildausgaben mit einem Schieberegister (2 × D 195 D) und das Synchronisieren von Schiebetakt und Programmablauf erfordern spezielle Hardware. Der Zeichengenerator[21] besteht aus einer Tabelle im Betriebssystem, die Teil des ersten EPROM ist (%ØF4Ø bis %ØFFF). Im Vergleich zu Lösungen mit einem speziellen EPROM als Zeichengenerator verringert sich so die Anzahl nötiger Bauelemente. Dafür wird für jeden Bildpunkt ein Bit im RAM benötigt statt nur eines Bytes je Zeichen. Der höhere Bedarf (1/2 Kbyte graphischer Bildwiederholspeicher)[22] läßt jedoch uneingeschränkt graphische Bildschirmausgaben zu.

Die Synchronität von Schiebetakt und Programmablauf gewährleistet die Taktzentrale (**Abb. 6**). Zwei Gatter des DL 000 D erzeugen den Systemtakt des Einchipmikrorechners von 8 MHz. Der 1:4-Teiler (DL 074 D) leitet daraus den Schiebetakt (2 MHz) ab. Das Bildausgabeprogramm ist so organisiert, daß alle 32 Systemtakte (acht Schiebetakte) ein Byte aus dem graphischen Bildwiederholspeicher gelesen wird. Die Taktlogik erkennt diesen Zeitpunkt durch 1-Pegel an den Bits PØ1 bis PØ7 (A9 bis A15, höhere sieben Adreßbits) und aktives $\overline{DS}$-Signal mit dem DL 030 D. Er erzeugt das Signal MC des Schieberegisters[23] zur parallelen Übernahme der Daten aus dem RAM über den Datenbus DØ bis D7 in das Schieberegister und das Synchronisieren der Phasenlage des Taktteilers (RC-Kombination).

Liest der Prozessor nicht aus diesem Speicherbereich, wird das Schieberegister über den seriellen Eingang ES mit 1-Pegel geladen, was der Dunkeltastung des Bildes entspricht. Der mit diesem Prinzip verbundene Hardwareaufwand ist kaum noch zu unterbieten. Es hat aber den Schönheitsfehler, daß außerhalb des für die Ausgabe genutzten Bildschirmbereichs weiße Striche entstehen, wenn der Prozessor den Bildinhalt ändert. Die beiden Dioden mischen das Helligkeitssignal aus dem Schieberegister mit den Synchronimpulsen, die der Einchip-Mikrorechner über das Port-3-Signal P37 ausgibt.

Der Tastaturanschluß erfolgt mit Hilfe der vom DS 8212 D erzeugten Adreßbits AØ bis A3 (Abb. 4). Die Eingabe über den Datenbus (D4 bis D7) realisiert der Tristate-Treiber U 40098 D. Er wird mit dem Auswahlsignal $\overline{CS3}$ immer dann freigegeben, wenn eine Adresse zwischen %6ØØØ und %7FFF vorliegt. Die Tastatur fragt der Prozessor durch Lesen der Speicherzellen %7FØ1 bis %7FØF ab. So lassen sich bis zu 15 × 4 Tasten anschließen. Die Zuordnung von Zeichencodes (ASCII) zu den einzelnen Tasten geschieht über eine Tabelle im EPROM1 (%ØFØØ bis %ØF3F).

Wie fertigen wir die Leiterplatte?

Die Grundlage bildet ein 135 mm × 155 mm großes Stück

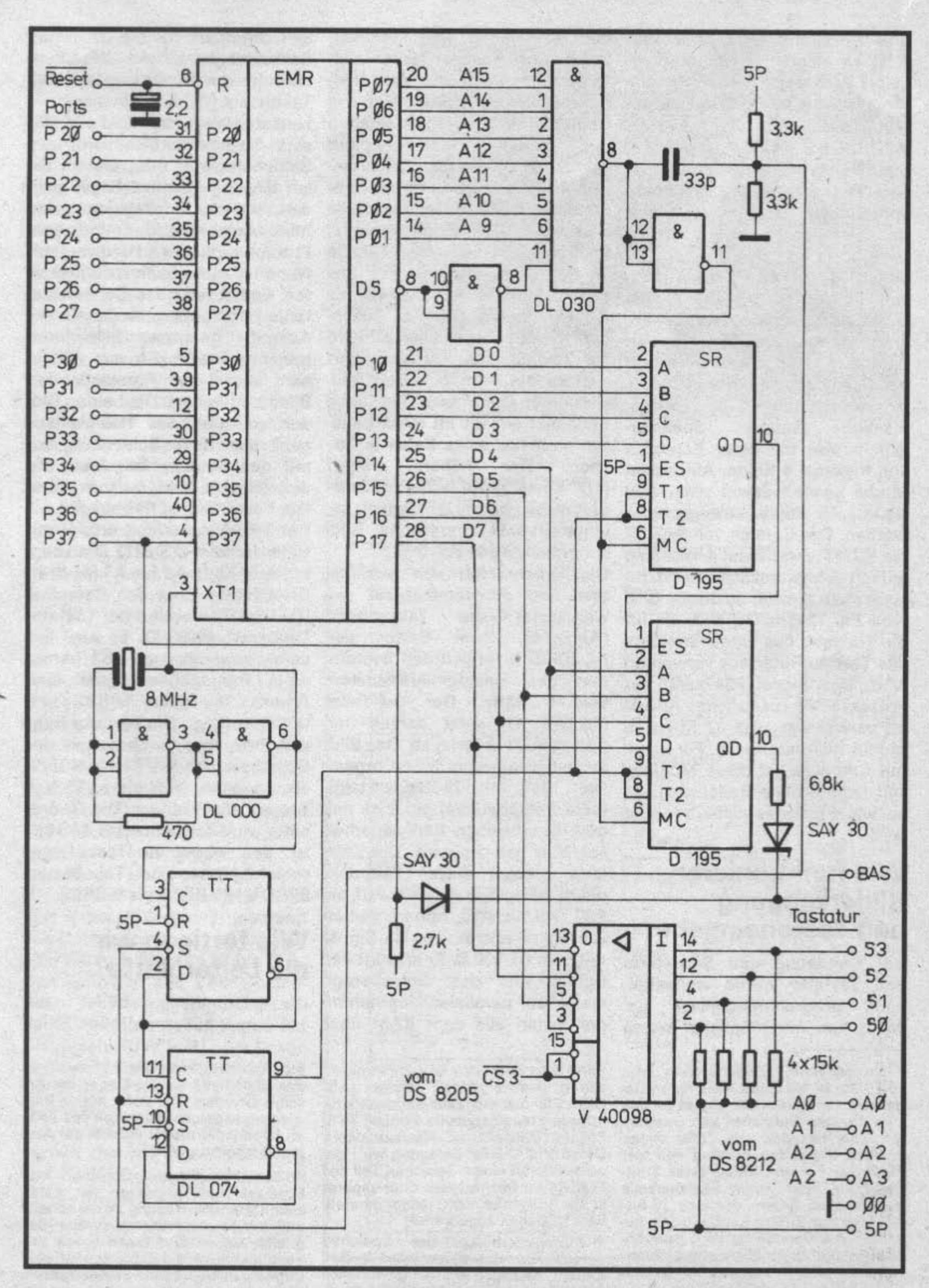
---

[20] **hexadezimal** Zahlensystem mit 16 Ziffern (Ø bis 9 und A bis F). Im Gegensatz zum dezimalen eignet es sich zum direkten Darstellen von Dualzahlen. Eine hexadezimale Ziffer widerspiegelt dabei den Zustand von vier **Flipflops**. Für den Inhalt eines 8-bit-Registers oder einer Speicherzelle werden zwei Ziffern, für eine 16-Bit-Adresse vier Ziffern benötigt. Die hexadezimale Darstellung wird gewöhnlich mit dem Präfix % gekennzeichnet. Die Zahl %231F bedeutet im Dualen ØØ1Ø ØØ11 ØØØ1 1111.

[21] **Zeichengenerator** Speicherbereich zum Zuordnen von Punktmatritzen zu anzuzeigenden Zeichen (**ASCII**). Die lesbare Zeichendarstellung erfordert mindestens 5×4 (meist 7×5) Punkte (Bildschirm, Matrixdrucker). Deren Hell-Dunkel-Belegung wird gewöhnlich mit einem Speicher, Teil des EPROM im Rechnerkern oder separat in der Ausgabeeinheit, jedem darstellbaren Zeichen zugeordnet.

[22] **Bildwiederholspeicher** Speicherbereich, der das darzustellende Bild enthält. Anzeigen werden gewöhnlich ständig wiederholt angesteuert (z. B. Fernsehbild oder 7-Segment-Multiplex-Tableau), was das Aufbewahren des Bildinhalts für die Dauer der Anzeige erfordert. Der dafür nötige Bildwiederholspeicher kann Teil des RAM im Rechnerkern oder separat der Ausgabeeinheit zugeordnet sein.

[23] **Schieberegister** Gruppe von **Flipflops** zum Anpassen der parallelen Datenübertragung an die serielle und umgekehrt. Sie gestatten das prallele Laden und Lesen sowie das Verschieben des Inhalts um je eine Bitposition bei jedem Schiebetaktimpuls.



6 Beschaltung des Einchip-Mikrorechners mit Taktzentrale, Schieberegister und Eingabetreiber

beidseitig kupferkaschierten Basismaterials. Am besten verarbeitet sich glasfaserverstärktes Kunstharz (Cevausit). Um Übereinstimmung von Löt- und Bestückungsseite zu erreichen, sollte vor dem Ätzen gebohrt werden. Die Körnpunkte lassen sich durch die im Maßstab 1:1 gedruckte Leiterbildzeichnung der Lötseite **(vgl. Abb. IV. Umschlagseite)** mit einer spitzen Reißnadel auf das fixiert untergelegte Basismaterial durchdrükken.

Vor dem Bohren müssen wir beide Seiten mit einem Scheuermittel metallisch blank reiben. Bereits hier sollte man große Sorgfalt walten lassen, um Schwierigkeiten beim Auftragen des Abdecklacks zu vermeiden. Gebohrt wird am besten mit einer Ständerbohrmaschine. Verfehlte und ungenau durchgedrückte Körnpunkte erschweren das spätere Bestücken, also auch hier konzentriert zu Werke gehen! Nach dem Bohren wird nicht entgratet, damit beim Auftragen des Abdecklacks nichts durch die Bohrlöcher auf die andere Seite gelangt.

Das Zeichnen der Leiterzüge und das anschließende Ätzen erfordern eine absolut fettfreie Oberfläche. Sie läßt sich mit Seife und Bürste erzielen. Danach ist das Berühren der Kupferflächen unbedingt zu vermeiden!

Während das entfettete und mit klarem Wasser abgespülte Basismaterial trocknet, bereiten wir den Ätzschutzlack vor. Zunächst empfiehlt sich das Einfärben mit Kopierstiftabrieb, um beim Zeichnen eine optische Kontrolle zu haben. Dazu füllen wir am besten ein Tintenfäßchen halb mit Abdecklack, schleifen mit feinem Sandpapier etwas Kopierstift ab und rühren das Pulver unter. Danach kann durch Verdünnen mit Spiritus die erforderliche Fließfähigkeit des Lacks erzielt werden. Günstigerweise machen wir hierzu Proben auf ein Stück Abfallmaterial. Der Lack muß leicht von der Zeichenfeder auf die Kupferfläche fließen, darf aber nicht breitlaufen. Als Ätzschutzmaske reicht schon eine sehr dünne Lackschicht, deren Oberfläche aber nicht angekratzt werden darf. Je besser man die optimale Konsistenz findet, desto leichter fällt das Zeichnen.

Das Beginnen mit der Lötseite empfiehlt sich wegen der guten Übersicht. Indem wir zunächst den 29-poligen Systembus zeichnen, können wir am besten die Fertigkeit zu 0,5 mm starken Leiterzügen und Lötaugenrändern üben. Wer geschickt genug ist, frei Hand zu zeichnen, hat einige Vorteile. Er kann sich besser auf das fehlerfreie Kopieren des Leiterbildes konzentrieren und läuft nicht Gefahr, mit dem Lineal bereits Gezeichnetes zu verwischen oder zu beschädigen. Ruhe und Konzentration sind geboten, denn jeder Zeichenfehler, jede unbedachte Bewegung, kann die spätere Inbetriebnahme erheblich verkomplizieren.

Während die fertig gezeichnete Lötseite trocknet, sollte man sein Werk einer gestrengen Kontrolle unterziehen. Fehlerhafte und fehlende Verbindungen lassen sich jetzt noch recht leicht korrigieren. Beim Ergänzen von Leiterzügen ist darauf zu achten, daß die Zeichenfeder den trockenen Lack nicht mehr aufkratzt. Überschüssigen Lack vorsichtig im trockenen Zustand mit der spitzen Reißnadel entfernen. Feuchter Lack würde nur verschmieren.

Ist die Lötseite nach etwa einer Stunde völlig trocken, drehen wir die Platte behutsam um. Der Schutz der Lötseite erfordert jetzt eine weiche Unterlage. Eventuell durchgelaufene Lackreste entfernen wir mit Reißnadel, Glaspinsel oder Rasierklinge gründlich, ohne dabei auf das Kupfer zu fassen. Um die richtigen Lötaugen zu treffen, ist es beim Zeichnen der Bestückungsseite **(vgl. Abb. III. Umschlagseite)** wichtig, sich klarzumachen, welche Anschlüsse der in Arbeit befindliche Leiterzug untereinander verbindet. Das Eindringen von Lack in die Bohrlöcher müssen wir zum Schutz der Maske auf der Lötseite völlig vermeiden. Dabei helfen die auf der Bestükkungsseite höheren Grate. Das Bedecken von größeren zusammenhängenden Flächen ist zwar mühselig, darf aber nicht überstürzt werden. Eine unbedachte Bewegung kann alle bisherige Mühe zunichte machen.

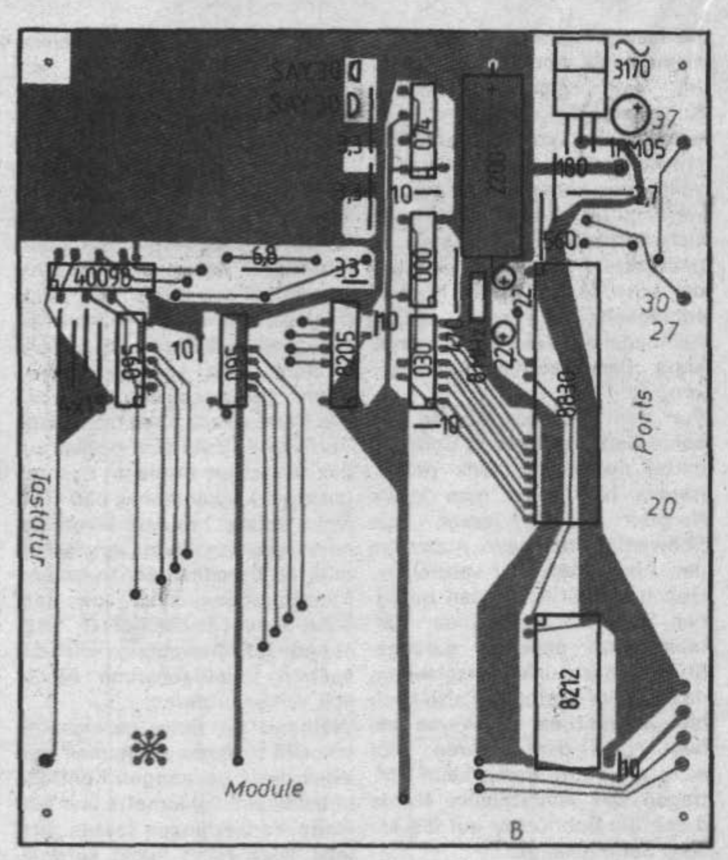
Nach Abtrocknen, Kontrolle und ggf. Korrektur folgt das Ätzen. Beim Ansetzen der Lösung verfahren wir nach der Gebrauchsanweisung. Vor und während des Ätzens darf die Lackmaske nicht beschädigt werden. Das gewährleisten am besten isolierte Drahtschlaufen, die wir durch die Befestigungsbohrungen an den Ecken ziehen. Sie gestatten die Handhabung ohne Berühren der Platine. Vorsichtiges Bewegen im Bad beeinflußt das Ätzen positiv. Wenn alles nicht abgedeckte Kupfer aufgelöst ist, wird die Platine unter fließendem kalten Wasser gründlich gespült. Keinesfalls darf die Platte zu lange im Ätzbad verweilen.

Zum Entfernen der Lackmaske empfiehlt sich das Abpinseln mit Spiritus. Anschließend sollten wir die Platine mit Kolophoniumlösung bestreichen, um die Lötbarkeit zu verbessern. Hier eignet sich Azeton am besten als Lösungsmittel, denn Spiritus ergibt eine unangenehm klebrige Schicht.

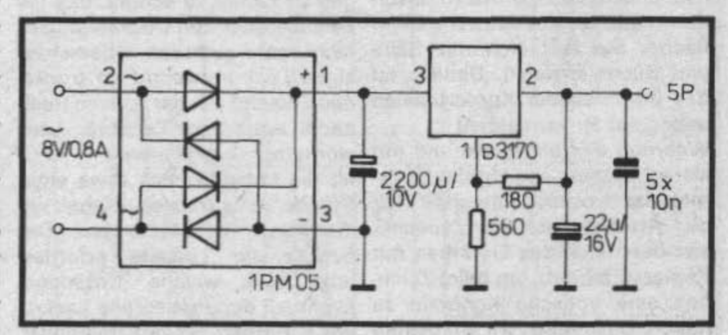
Wann setzen wir die Bauelemente ein?

Nach Trocknen der Kolophoniumschicht löten wir als erstes die freien Durchkontakte ein. Sie sind in der Bestückungszeichnung **(Abb. 7)** mit kleinen Kreisen markiert. Am besten eignet sich hierfür lötfähiger Draht mit einer Stärke, die knapp in die Bohrlöcher paßt. Er sollte bei jedem Durchkontakt mit einer größeren Kombizange oder einer kleinen Wasserpumpenzange breitgedrückt werden, so daß er nicht mehr herausfallen kann. Dieses Verfahren gewährleistet am besten, daß beim anschließenden Verlöten kein Durchkontakt ver-

sehentlich mit dem Lötkolben wieder herausgezogen wird.
Beim Einsetzen der Schaltkreise darf keine falsche Positionierung unterlaufen. Es empfiehlt sich, vor dem Löten noch einmal Typ und Lage des Anschlusses 1 (Gehäusemarkierung) zu überprüfen. Begonnen wird mit dem U 40098 D, weil später dessen Lötstellen auf der Bestückungsseite schwer zugänglich sind. Sparsamer Zinnverbrauch beugt ungewollten Kontakten zwischen benachbarten Anschlüssen vor. Am einfachsten ist, zunächst die eingesetzten Schaltkreise auf der Bestückungsseite anzulöten und dann die Verbindungen auf der Lötseite herzustellen.
Es folgt das Bestücken der Widerstände. Die vier Pull-up[24] des U 40098 D (15k) und der 180-Ω-Widerstand müssen einseitig auch auf der Bestückungsseite verlötet werden. Alle anderen Anschlüsse dürfen nur auf der Lötseite Kontakt bekommen. Die Stützkondensatoren am DS 8212 D und am D 195 D werden ebenfalls einseitig auch auf der Bestückungsseite angelötet. Für die spätere Inbetriebnahme ist es günstig, statt des 33-pF-Kondensators einen Trimmer 10/40 und statt des 6,8-k-Widerstandes einen Einstellregler 22k einzusetzen. Nach Abgleich auf optimale Bilderzeugung ersetzen wir sie später durch Festwerte.
Vor dem Bestücken des Ladeelkos (2200 µF/10 V) überprüfen wir, ob er am positiven Anschluß Kontakt mit der Bestückungsseite bekommt. In diesem Fall muß mit einem größeren Bohrer (ca. 3 mm) freigesenkt werden. Der negative Anschluß realisiert einen Durchkontakt. Beim Löten auf der Bestückungsseite empfiehlt sich minimaler Zinneinsatz, um den Raum für den Quarz und den 22-µF-Elko nicht unnötig einzuengen. Es folgt das Bestükken der beiden übrigen Elkos, der



7 Bestückungsplan der Prozessorplatine



8 Schaltbild der Stromversorgung

Dioden und des Schwingquarzes. Statt des Quarzes kann zur Not auch ein 82-pF-Kondensator verwendet werden, was aber eine schlechtere Frequenzkonstanz zur Folge hat.
Der Spannungsregler B 3170 V benötigt eine Kühlfläche von wenigstens 50 mm². Deshalb sollte er erst nach der Montage an eine geeignete Konstruktion eingelötet werden. Seine Anschlüsse dürfen nur mit einer Rundzange gerichtet werden, da sonst die Gefahr des Abbrechens besteht. Die Kühlschelle liegt auf 5-V-Potential und ist deshalb isoliert am Kühlkörper zu befestigen. Günstig wirkt sich das Schwärzen der nach außen gerichteten Kühlflä-

[24] Pull-up Widerstand oder Transistor zum Erzeugen des 1-Pegels eines Signals (im Gegensatz zu pull-down: Ø-Pegel)

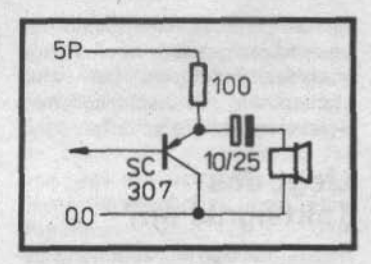
chen mit einer sehr dünnen Lackschicht aus, da so mehr Wärmeenergie abgestrahlt wird.
Haben wir einen UB 8831 D verwendet, müssen wir eine Lötbrücke zwischen dessen Pins 1 und 2 ergänzen. Das sichert die Betriebsspannungsversorgung des internen RAM. Beim UB 8830 D bleibt Pin 2 (Ausgang des nicht benutzten internen Taktgenerators) frei.

Was funktioniert denn schon alles?

Zur Inbetriebnahme der Prozessorplatine nutzen wir den Test-Modus des Einchip-Mikrorechners. Dazu muß eine Drahtbrücke die Port-Anschlüsse P32 und P35 verbinden. Sie sind über die rechts vom Einchiprechner-Schaltkreis angeordneten Lötstützpunkte (vgl. Abb. III. Umschlagseite) zugänglich. Deren Anordnung entspricht von unten nach oben der Numerierungsfolge in der Anschlußbezeichnung (P2Ø bis P37). Außerdem ist zwischen dem Lötstützpunkt am 2,2-µF-Elko und Masse eine Rücksetztaste (RESET) anzuschließen.
Zunächst nehmen wir die Stromversorgung in Betrieb, deren Schaltung in **Abb. 8** dargestellt ist. Sie benötigt einen Netztrafo mit 8 V Sekundärspannung bei 0,8 A (nur Grundgerät) bzw. 1,1 A (mit EPROM-Programmierzusatz, vgl. Teil II) Laststromstärke. Ideal ist ein Klingeltrafo, da er die erforderliche Schutzgüte bereits gewährleistet. Bei Einsatz eines anderen Trafos (z. B Heiztrafo 8,3 V/1,3 A) muß mit einem Schuko-Anschluß und vorgeschalteter Sicherung die Schutzart Nullung realisiert werden. Hier ist die **Abnahme des Trafo-Anschlusses durch einen Fachmann** vor dem ersten Einschalten erforderlich. Haben wir nur einen 12-V-Trafo vorrätig, muß der Ladeelko auf 1000 µF/16 V geändert und der Spannungsregler B 3170 V stärker gekühlt werden.
Verdrahtungsarbeiten erfolgen stets bei abgeschalteter Netzversorgung. Zunächst schließen wir den Trafo über ein Amperemeter an die beiden mit dem Wechselspannungszeichen markierten Lötstützpunkte rechts oben am Gleichrichter 1PM05 an. Es empfiehlt sich, nochmals die richtige Position aller Schaltkreise und des Gleichrichters anhand des Bestückungsplanes (Abb. 9) zu überprüfen, bevor wir den Trafo mit dem Netz verbinden.
Die typische Stromaufnahme der Prozessorplatine beträgt etwa 600 mA (Wechselstrom). Das entspricht einer Gleichstromstärke von 420 mA. Fließt nach dem Einschalten ein erheblich stärkerer Strom (z. B. 700 mA), liegt ein Kurzschluß oder ein verdreht eingesetzter Schaltkreis vor. Deshalb sofort wieder abschalten. Läßt sich keine Ursache finden, kann auch eine zu hohe Ausgangsspannung des B 3170 V vermutet werden. Sie entsteht durch zu kleinen Wert des 180-Ω- oder zu großen des 560-Ω-Widerstandes. An den vier Lötstützpunkten rechts neben dem DS 8212 D sind von oben nach unten 5P (+5 V), Masse (00) und die beiden Sonderspannungen (U1 und U2) zugänglich. Hier kann für die folgenden Messungen das Bezugspotential (00) gewonnen werden. Wir müssen sehr vorsichtig vorgehen, da auch beim Messen ein Kurzschluß zum Zerstören von Bauelementen führen kann. Liegt die Spannung am 5P-Anschluß außerhalb des Bereiches von 4,9 bis 5,1 V, arbeitet der Spannungsregler nicht ordnungsgemäß. Durch Variieren des 180-Ω- oder des 560-Ω-Widerstandes, erfolgt ggf. die Korrektur.
Am Pin 1 des Einchip-Mikrorechners und am Pin 24 des DS 8212 D muß das 5-V-Potential anliegen. Bei den übrigen Schaltkreisen ist es an den Anschlüssen 14 bzw. 16 (gegenüber Pin 1) zu überprüfen. Anschließend können die Masseanschlüsse (UB 8830 D: Pin 11, DS 8212 D: Pin 12, sonst: Pin 7 bzw. Pin 8) kontrolliert werden. An ihnen dürfen höchstens 10 mV anliegen. Höhere Spannungsabfälle deuten auf die Funktionssicherheit beeinträchtigende hochohmige Masseverbindungen hin. Hier können wir mit nachträglichem Verzinnen Abhilfe schaffen.

Liegt das Taktsignal an?

Neben der Betriebsspannung benötigt der Prozessor das Taktsignal. Am Pin 3 des UB 8830 D sowie an den Pins 8 und 9 der beiden D 195 D mißt man bei funktionierender Taktversorgung mit dem Voltmeter etwa 2 V. Mit einem Oszilloskop können wir auch die Frequenz von 8 MHz (am Prozessor) bzw. 2 MHz (Schiebetakt) kontrollieren. Am Pin 1 des DS 8212 liegt bei fehlerfreier Funktion ebenfalls eine Schwingung mit unregelmäßiger Folge (ca. 1 MHz) vor. Läßt sie sich nicht nachweisen, kann man es noch einmal am Pin 9 des UB 8830 D versuchen. Hat man nur hier Erfolg, ist der betreffende Leiterzug (unter dem Prozessor) unterbrochen. Übrigens lassen sich Unterbrechungen an Hand des Schaltbildes (Abb. 4 und 6) allgemein so am besten aufspüren.
Tritt am Pin 9 des Prozessors auch nach Betätigen der Rücksetztaste trotz funktionierender Taktzentrale keine Schwingung auf, ist eine Verbindung dieses Pins mit Masse oder 5P eine mögliche Ursache. Mitunter genügt dem Prozessor nicht der hohe Pegel des Systemtaktes. Das läßt sich mit einem Widerstand (ca. 1k) zwischen Pin 6 des DL 000 D und 5P „geradebiegen".
Ist der Prozessor in Betrieb, wird der 29-polige Systembus geprüft. Steht kein Oszilloskop zur Verfügung, kann man sich mit einem Piezoschwinger (z. B. aus einer Armbanduhr) oder Kopfhörer höherer Impedanz (mindestens 1k) zwischen 5P und dem zu überprüfenden Signal behelfen. Für Kopfhörer niedrigerer Impedanz, Telefonhörkapseln und Lautsprecher ist ein Verstärker **(Abb. 9)** erforderlich, um die Schaltkreis-Ausgänge nicht zu überlasten.
Bei fehlerfreier Funktion muß an



9 Abhörverstärker für die Inbetriebnahme

den miteinander verbundenen Signalen P32 und P35 eine 5,45-kHz-Schwingung erkennbar sein. Die Impulsbreite beträgt nur 2,5 µs, so daß ein relativ leiser Ton entsteht. Beginnt diese Schwingung auch nicht nach Betätigen der Rüchsetztaste, ist mindestens ein Bit der Signale PØØ bis PØ7 oder P1Ø bis P17 (Anschlüsse 13 bis 28 des UB 8830 D) falsch beschaltet. Sie müssen ggf. auf Fehlverbindungen bis zu den Schaltkreisen DS 8212 D, DL 030D, DS 8205 D, D 195 D und U 40098 D untersucht werden. Sie dürfen nur mit den in der Schaltung angegebenen Anschlüssen, aber keinesfalls mit Masse, 5P oder untereinander Verbindung haben.

Wenn die Signale PØØ bis P17 alle mit 5,45 kHz schwingen, führt der Prozessor das Testprogramm aus. Hier entsteht wegen der Impulsbreite von 11 µs ein lauterer Ton als an P35. Bei korrekter Taktfrequenz liegt er zwischen e^5 und f^5, was sich bei musikalisch geschultem Gehör mit Hilfe eines Musikinstrumentes kontrollieren läßt. Um Leiterzugunterbrechungen auszuschließen, müssen die Töne auch an den entlegenen Stellen, an den genannten Schaltkreisen, überprüft werden. Wenn der DS 8212 D funktioniert, verfolgen wir an allen Bussignalen zwischen Masse (breiter Leiterzug) und R/W der Modulsteckplätze diese Schwingungen. Beim Messen kann durch kurzzeitige ungewollte Verbindungen das Testprogramm gestoppt werden. Mit wiederholtem Rücksetzen ($\overline{\text{RESET}}$-Taste) läßt es sich dann wieder starten.

Als letztes können wir die Steuersignale überprüfen. Sie schwingen (mit Ausnahme von $\overline{\text{AS}}$) beim Testbetrieb nicht. R/W und alle acht Auswahlsignale ($\overline{\text{CSA}}$, $\overline{\text{CSB}}$) an den oberen Enden der Modulsteckplätze (vgl. Abb. IV. Umschlagseite) müssen ständig 1-Pegel (ca. 3,5 V) führen. Auch die Pins 1 und 15 des U 40098 D liegen auf 1, während die Pins 6 der beiden D 195 D mit Ø (höchstens 0,5 V) belegt sind. Treten hiervon Abweichungen auf, müssen wir die Signale ab $\overline{\text{DS}}$ (Pin 8 des UB 8830 D) verfolgen, bis eine Unterbrechung oder ein Schluß gefunden ist.

Bei Ausgangsüberlastungen gehen die Schaltkreise meist nicht sofort, aber mit der Zeit kaputt. Daher dürfen keinesfalls bei der Suche nach Fehlern Dauertests durchgeführt werden. Erst wenn alle genannten Signale die richtige Schwingung bzw. den richtigen Pegel aufweisen, ist die Prozessorplatine zumindest für den Speicheranschluß betriebsfähig.

3. Speichermodule

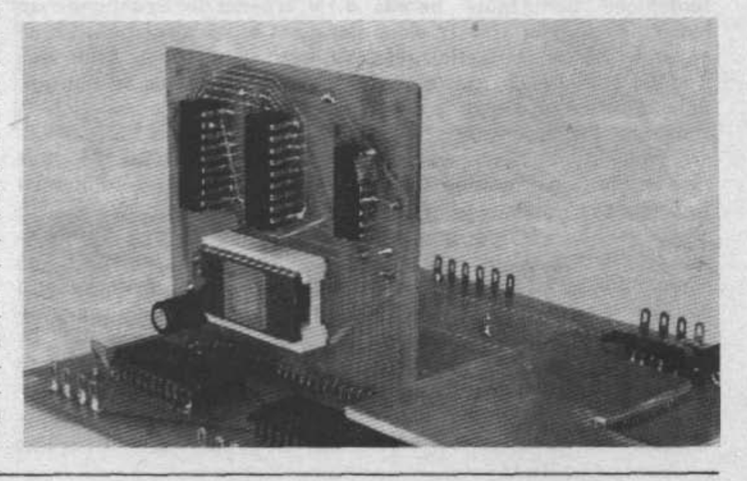
Welche Speicherschaltkreise eignen sich?

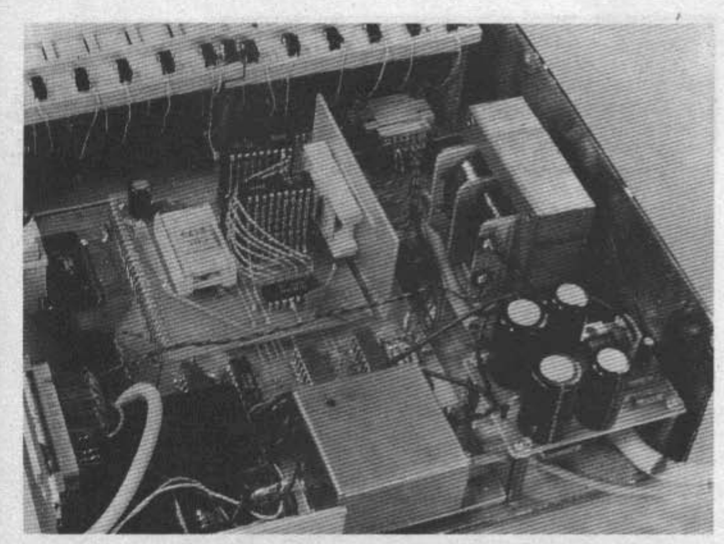
Unser Computer braucht zu zwei Zwecken zwei verschiedene Arten von Speichern. Das sind einerseits Festwertspeicher für Programme, die nach dem Einschalten sofort zur Verfügung stehen müssen, und Code-Tabellen, andererseits operative Speicher, wo variable Informationen wie Anwenderprogramme und Bildinhalte Platz finden. Beide Eigenschaften gleichzeitig lassen sich mit batteriegestützten CMOS-RAM-Schaltkreisen erzielen. Dennoch ist es günstiger, für das Betriebssystem Festwertspeicher (EPROM) zu verwenden. So ist das Installieren des Betriebssystems leicht und dessen Verlust kaum möglich.

Der Systembus läßt alle in der DDR gefertigten EPROM ab 1 K × 8 bit zu. Jedoch erfordert der Einsatz des U 555 C, der diese Kapazität besitzt, erheblichen Zusatzaufwand, so daß wir den U 2716 C (2 K × 8 bit) verwenden. Wir stellen auch Modulplatinen vor, die neben dem U 2716 C ohne Änderungen an der Hardware die Typen U 2732 C (4 K × 8 bit) und U 2764 C (8 K × 8 bit) zulassen. Unser Betriebssystem-Listing (vgl. Seiten 24/25) ist jedoch auf zwei 2-K-EPROM zugeschnitten, da diese sich am leichtesten beschaffen lassen.

Für den operativen Speicher sind RAM-Schaltkreise (Schreib-Lese-Speicher) nötig. Da unser Com-

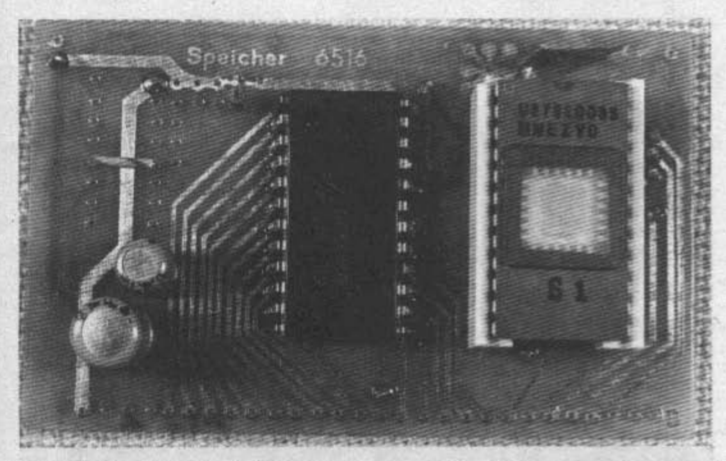
Aufgesetzter Speichermodul (Variante A) mit 2 Kbyte RAM





Der geöffnete Computer mit „Schaltkreistürmchen" zum Realisieren von 8 Kbyte RAM (Basis: U 224 D)

Speichermodul Variante B mit einem U 6516 D (2 Kbyte RAM)



puter keine Auffrisch-Zyklen (refresh) ausführt, können keine dynamischen RAM-Schaltkreise verwendet werden. Unter den statischen RAM sind die in CMOS-Technologie gefertigten vorzuziehen, da sie weniger Betriebsstrom benötigen und das Ergänzen einer Stützbatterie zum Datenerhalt auch bei ausgeschaltetem Computer gestatten. Den RAM können wir wahlweise mit den Typen U 224 D (1 K × 8 bit, deshalb mindestens vier Stück), U 6516 D oder U 6264 D aufbauen. Die beiden letztgenannten bieten dank ihrer Organisation (2 K × 8 bit bzw. 8 K × 8 bit) bereits mit einem Schaltkreis ausreichende Kapazität.

Bastlertypen der Kategorie S1 führen nur ganz selten zu Ausfällen. Schwierigkeiten bereiten dagegen langsame EPROM (ab 400 ns Zugriffszeit). Setzen wir solche Schaltkreise ein, wie z. B. den U 2716 C 45, müssen wir den $\overline{CS}$-Anschluß (Pin 18) vom Auswahlsignal trennen und mit Masse verbinden. Das Auswahlsignal vom DS 8205 D der Prozessorplatine steuert den EPROM dann nur noch mit dem $\overline{OE}$-Eingang (Pin 20). Dadurch verzichten wir aber auf den energiesparenden Stand-by-Zustand[25]. Die im folgenden vorgestellten Leiterplatten der Speichermodule besitzen alle eine Topologie (Leiterzugführung), die das Durchtrennen des Leiterzuges direkt am Lötauge des $\overline{CS}$-Eingangs gestattet, ohne die Beschaltung des $\overline{OE}$-Eingangs zu verändern. Das so isolierte Pin 18 kann dann durch einen kurzen Schaltdraht mit Masse (Pin 12) verbunden werden.

Wie findet der Prozessor Speicherinhalte?

Der Einchip-Mikrorechner greift immer nur auf eine einzelne Speicherzelle zu. Um zwischen den vielen tausend möglichen auszuwählen, gibt er zunächst deren Adresse aus. Das sind, wie wir wissen, 16 Bit, deren untere acht (AØ bis A7) vom DS 8212 D für die Dauer eines Zugreiffs (Zyklus) gespeichert werden. Diese und die nächsten fünf (A8 bis A12) sind Teil des Systembus (vgl. Abb. 4). Alle Speicherschaltkreise sortieren im Innern die so bestimmte Zelle heraus. Der Prozessor muß nun auch noch unter den Schaltkreisen wählen. Das macht er mit dem DS 8205 D, der das aktivierende $\overline{DS}$-Signal an nur einen der angeschlossenen Speicherschaltkreise weitergibt. Hier wird, je nach Belegung von R/W (Lesen/Schreiben), der Inhalt der betreffenden Zelle auf die Datenbits DØ bis D7 geschaltet oder von dort in die Speicherzelle übernommen und bis auf Widerruf aufbewahrt. Die anderen, nicht aktivierten Speicherschaltkreise bleiben völlig passiv. Mit AØ bis A12 lassen sich 8192 = 8 K Speicherzellen unterscheiden. Soll dieser Bereich durch mehrere Schaltkreise geringerer

[25] **Stand-by-Zustand** Passive Bereitstellung. Gewöhnlich enthält ein Rechner mehrere Speicher-Schaltkreise (**EPROM, RAM**), von denen nur jeweils einer aktiv ist. Einige Schaltkreistypen lassen sich zu den anderen Zeiten teilweise passivieren, um Strom zu sparen.

Kapazität abgedeckt werden, ist ein weiterer DS 8205 D nötig, der abhängig von A1Ø, A11 und A12 das Auswahlsignal vom DS 8205 D der Prozessorplatine weiter verteilt. Für die RAM-Schaltkreise der Typen U 224 D und U 6516 D enthalten die Module den entsprechenden Dekoder. Bei den EPROM wurde darauf verzichtet, da das volle Ausnutzen der Kapazität angesichts des geringen Bedarfs nicht wichtig ist und auf diese Weise sehr kleine Modul-Leiterplatten konstruiert werden konnten.

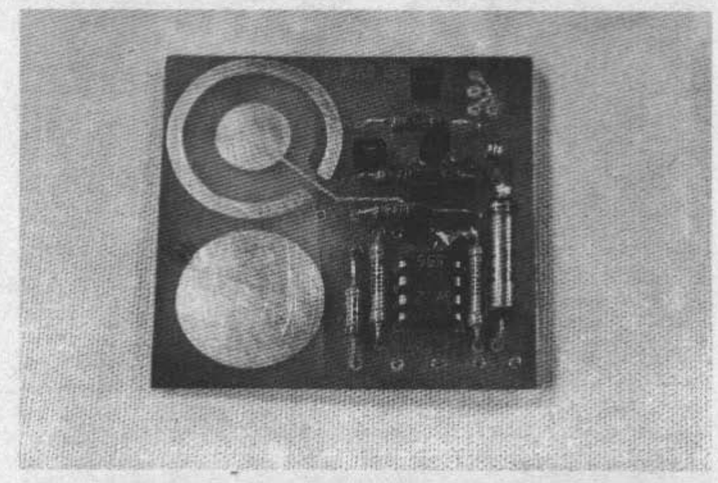
Den Adreßraum (vgl. Abb. 5) können wir mit RAM und EPROM beliebig nutzen. Die Konzeption des Betriebssystems (Programme zur Rechner-Verwaltung) sieht in der unteren Hälfte Festwert- (EPROM) und in der oberen Hälfte Operativspeicher (RAM) vor. So lassen sich bereits mit einem Modul und den beiden zugeordneten Auswahlsignalen $\overline{CSA}$ (RAM) und $\overline{CSB}$ (EPROM) beide Teile realisieren (vgl. Abb. 5).

Wegen der Verteilung des 4-K-Betriebssystems auf zwei EPROM U 2716 C ist ein zweites Modul erforderlich. Da die RAM-Kapazität eines Moduls mit 8 Kbyte für die meisten Anwendungen völlig ausreicht, benötigt der zweite keinen Operativspeicher. Hier kann die Modulvariante D zum Einsatz kommen (vgl. Abb. 23, 24 und 25). Grundsätzlich darf jede Modulvariante, die wir nachfolgend vorstellen, auf jedem Platz angewendet werden. Auf Platz 4 kann jedoch wegen des betreffenden Auswahlsignals ($\overline{CSB}$) für die Tasteneingabe kein EPROM sitzen. Der RAM-Bereich von Modul 1 ist wie in **Abb. 10** dargestellt eingeteilt.

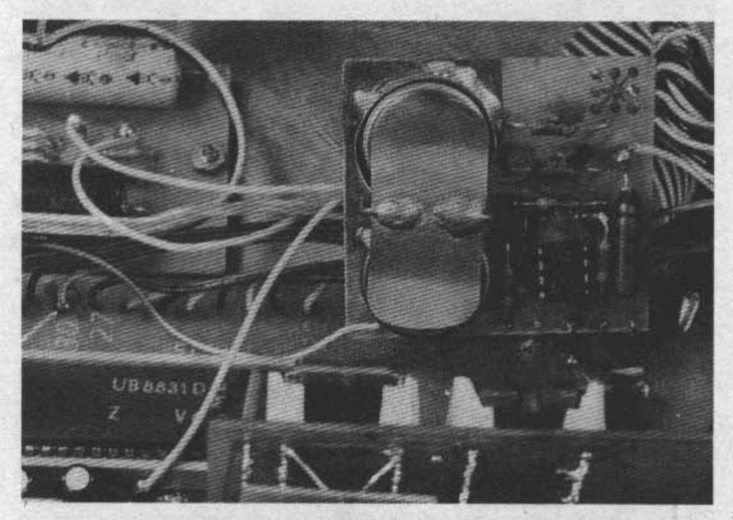
Außer dem Freibereich für Standardprogramme und dem Anwenderbereich werden diese Dateien vom Betriebssystem benutzt. Für den Anwender reicht auch oft schon 1 Kbyte, also der Bereich von % EØØØ bis % E3FF, so daß nur insgesamt 2 Kbyte RAM bestückt werden müssen: ein „Anwenderkilo" und ein „Systemkilo".

%FEØØ bis %FFFF: Graphik-Bildwiederholspeicher
%FD8Ø bis %FDFF: Stapelspeicher
%FDØØ bis %FD7F: ASCII-Bildwiederholspeicher
%FC8Ø bis %FCFF: frei für Standardprogramme
%FCØØ bis %FC7F: Magnetband-Puffer
%EØØØ bis %FBFF: Anwenderbereich

__10__ Aufteilung des RAM auf Modulplatz 1 für Betriebssystem und Anwenderprogramme



Die Batteriestütze für CMOS-RAM erfordert nur wenig Aufwand.

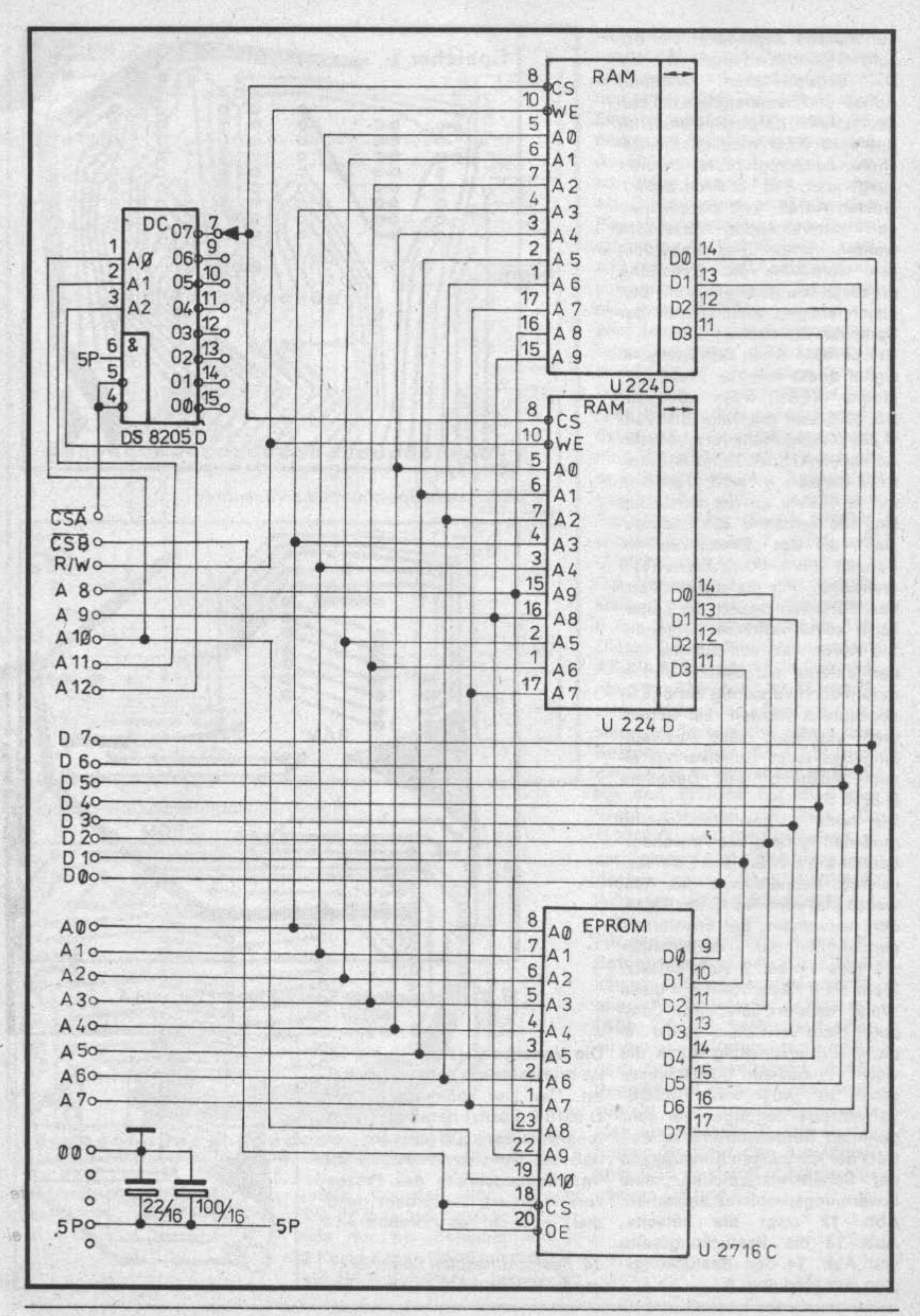


Batteriestütze mit Nickel-Cadmium-Zellen im eingebauten Zustand

Wie sehen die Speichermodule aus?

Die **Variante A** ist für die Typen U 2716 C und U 224 D ausgelegt. **Abb. 11** enthält ihr Schaltbild. Der RAM-Schaltkreis U 224 D ist 1 K × 4 bit organisiert, so daß für die in unserem Computer benötigte Verarbeitungsbreite von 8 bit zwei Stück eingesetzt wer-

__11__ Schaltbild der Modulvariante A mit Darstellung der unteren beiden U 224 D. Die weiteren werden parallelgeschaltet, nur der Anschluß $\overline{CS}$ wird mit einem anderen Ausgang des DS 8205 D verbunden.

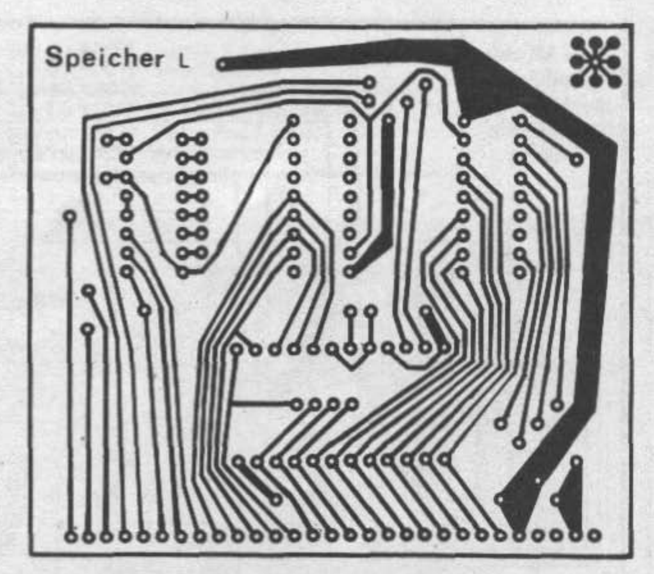


RAM
CS
WE
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
D0
D1
D2
D3
U224D
RAM
U 224 D
DC
DS 8205 D
&
5P
CSA
CSB
R/W
A 8
A 9
A 10
A 11
A 12
D 7
D 6
D 5
D 4
D 3
D 2
D 1
D 0
A0
A1
A2
A3
A4
A5
A6
A7
00
5P
22/16
100/16
EPROM
OE
D4
D5
D6
D7
U 2716 C

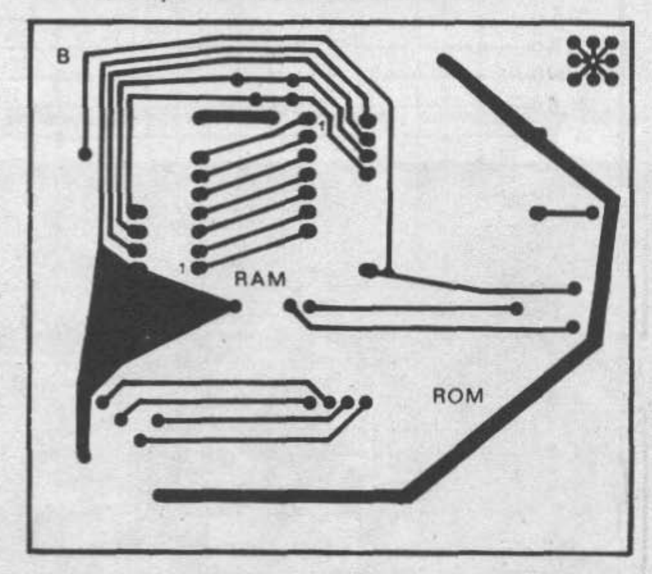
den müssen: Jeder ist an vier der acht Datenbusleitungen DØ bis D7 angeschlossen, während Adreß- und Steuersignale bei beiden parallel geschaltet sind. Im Interesse einer möglichst einfachen Leiterzugführung wurde ausgenutzt, daß bei RAM-Schaltkreisen Adreß- und Datenleitungen untereinander vertauscht werden dürfen. Dies eingerechnet, verbindet die Schaltung praktisch alle Bussignale mit den gleichnamigen Anschlüssen der Speicherschaltkreise.

Der EPROM erhält sein Auswahlsignal direkt von der Prozessorplatine ($\overline{CSB}$). Ein Dekoder DS 8205 teilt mit Hilfe der vom U 224 D nicht mehr verarbeiteten Adressen A1Ø, A11 und A12 den RAM-Bereich in acht Scheiben mit je 1Kbyte. In der Schaltung sind die beiden U 224 D dargestellt, die das „Systemkilo" im Bereich von % FCØØ bis % FFFF realisieren. Für die anderen sieben Scheiben besitzt die Leiterkarte keine weiteren Lötaugen. Sie werden einfach auf die beiden unteren aufgesetzt. Vorher muß jedoch vorsichtig Pin 8 ($\overline{CS}$) abgespreizt werden. Ein Schaltdraht verbindet diese Auswahl-Eingänge jeder Scheibe mit einem Ausgang des Dekoders D 8205 D.

Die erste aufgesetzte Scheibe muß den Anfang des Anwenderbereichs % EØØØ bis % E3FF realisieren. Deshalb wird das Auswahlsignal vom Pin 15 des Dekoders verwendet. Bei Erweiterungen kommen dann nacheinander die Pins 14 bis 9 zum Einsatz. Mehr als 8 Kbyte RAM auf diese Weise realisiert, kann unser Computer nicht verwalten, da die kapazitive Busbelastung durch die vielen parallelen Anschlüsse sonst zu groß wird. CMOS-Schaltkreise benötigen nur sehr geringen Betriebsstrom, so daß trotz der kompakten Konstruktion der Schaltkreistürmchen keine Erwärmungsprobleme entstehen. **Abb. 12** zeigt die Lötseite, **Abb. 13** die Bestückungsseite und **Abb. 14** den Bestückungsplan von Modultyp A.

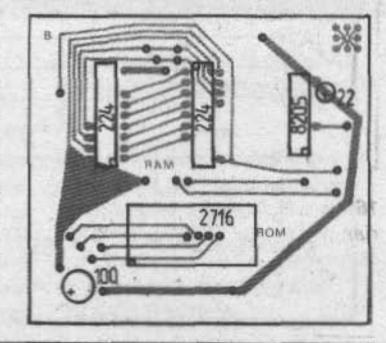


12 Lötseite Speichermodul Variante A

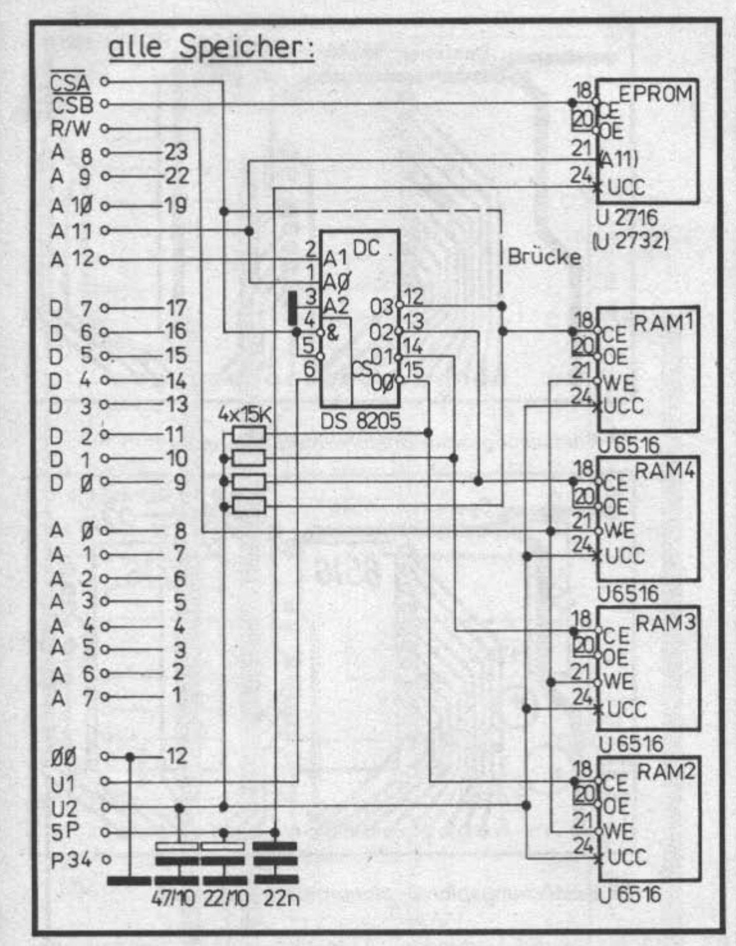


13 Bestückungsseite Speichermodul Variante A

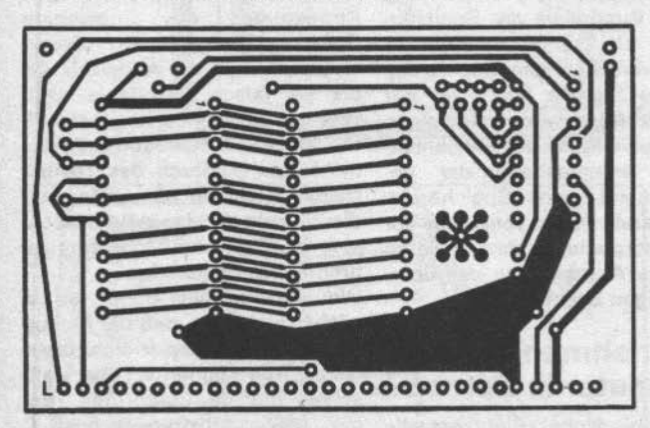
Die **Variante B (Abb. 15 bis 18)** ist grundsätzlich genauso gestaltet. Der hier verwendete RAM U 6516 D besitzt bereits acht Datenanschlüsse (DØ bis D7), so daß ein Schaltkreis bereits die Verarbeitungsbreite des Prozessors realisiert. Außerdem dekodiert er auch das Adreßbit A1Ø



14 Bestückungsplan Speichermodul Variante A



15 Schaltbild Speichermodul Variante B



16 Lötseite Speichermodul Variante B

intern, so daß der Dekoder DS 8205 D auf der Speicherplatine nur vier Scheiben mit je 2 Kbyte Kapazität bilden muß. Da ein RAM-Schaltkreis bereits für viele Fälle genügend Kapazität besitzt, kann auf den Dekoder auch verzichtet werden. Kommt nur ein U 6516 D (RAM 1) zum Einsatz, müssen statt des Dekoders eine Brücke zwischen den Anschlußpunkten 5 und 12 (gestrichelte Linie in Abb. 15) und ein Durchkontakt an Stelle des Pins 16 eingesetzt werden. Durch Ignorieren der Adreßbits A11 und A12 realisiert der U 6516 D dann sowohl das „System"- als auch ein „Anwenderkilo".

Zum Erweitern des Anwenderbereichs entfernen wir Brücke und Durchkontakt, bestücken einen DS 8205 D und setzen ein bis drei U 6516 D auf den schon vorhandenen. Die vorher abgespreizten Pins 18 und 20 erhalten durch einen Schaltdraht das Auswahlsignal vom DS 8205 D gemäß der Schaltung (Abb. 15). Als EPROM können die Typen U 2716 C und U 2732 C zum Einsatz kommen.

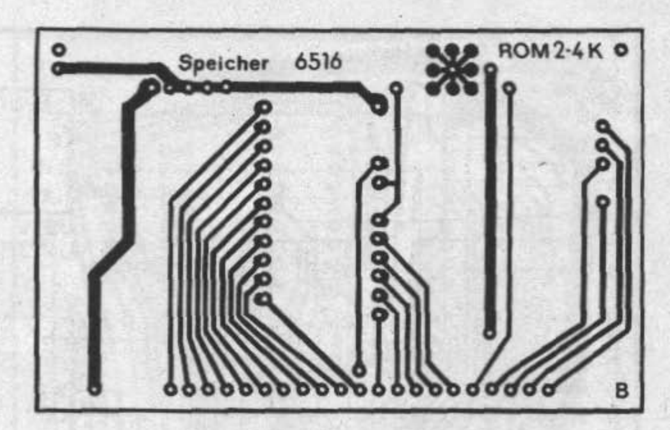
Die **Variante C** ist für Schaltkreise mit der Organisation 8 K × 8 bit ausgelegt **(Abb. 19).** Diese Typen dekodieren intern AØ bis A12, so daß kein Dekoder nötig ist. **Abbildung 20, 21 und 22** zeigen Löt- und Bestückungsseite der Leiterplatte sowie den Bestückungsplan. Als RAM muß ein U 6264 D verwendet werden, für den EPROM hat man die Wahl zwischen U 2716 C, U 2732 C und U 2764 C. Beim Einsatz eines U 27128 C wird nur dessen obere Bank (EPROM-Adressen %2ØØØ bis %3FFF) genutzt.

Für den zweiten EPROM des 4-K-Betriebssystems mit 2 U 2716 C können wir eine besonders einfache Modulvariante einsetzen **(Abb. 23).** Sie basiert auf einer einseitigen Leiterplatte mit mäßigem Schwierigkeitsgrad. **Abbildung 24 und 25** zeigen das Leiterzugbild und den Bestückungsplan der **Variante D.** Die Drahtbrücke Br führt das Auswahlsignal $\overline{CSB}$ zu den Schaltkreiseingängen $\overline{OE}$ und $\overline{CS}$.

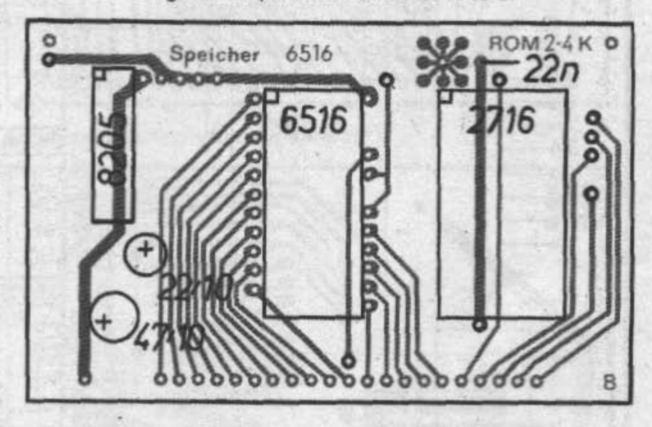
Wie fertigen wir die Module?

Wir fertigen die Leiterplatte wie bei der Prozessorplatine beschrieben. Bei den Modulvarianten B und C ist das Zeichnen der Lötseite sehr schwierig. Günstig ist hier, zunächst die Löcher vor dem Zeichnen mit möglichst geringem Durchmesser zu bohren (z. B. 0,5 mm) und nach dem Ätzen auf die richtige Größe (0,8 bis 1,0 mm) zu bringen. Die Leiterzüge zwischen den Lötaugen müssen mit einer Stärke von etwa 0,3 mm gezeichnet werden. Das erfordert das Führen der Zeichenfeder um 90° gedreht in senkrechter Haltung. Gewöhnlich gelingt das nicht auf Anhieb, so daß mit mehreren vergeblichen Versuchen gerechnet werden muß. Dem Einsetzen der Durchkontakte (kleine Kreise in den Bestückungsplänen) folgen die RAM-Schaltkreise. Ist mehr als eine Scheibe zu bestücken, sollte erst nach Aufsetzen der zweiten Ebene die Bestückungsseite gelötet werden. Jetzt sind die Stützkondensatoren und ggf. der Dekoder DS 8205 D an der Reihe. Für den EPROM setzen wir eine 24-polige, bei Modulvariante C eine 28-polige Schaltkreisfassung ein. Im Gegensatz zu den anderen Bauelementen dienen deshalb hier die Anschlüsse nicht als Durchkontakt.

Als nächstes werden 29 etwa 2 cm lange Drahtstücken, die z. B. beim Kürzen von Bauelemente-Anschlüssen anfallen, als Kontaktkamm in die Speicherplatine eingelötet. Dem gleichmäßigen Ausrichten folgt das Einfädeln in die zugehörigen Bohrungen des betreffenden Modulplatzes der Prozessorplatine. Platz 1 liegt dem DS 8212 D am nächsten, es folgen Platz 2, 3 und 4. Auf Platz 1 gehört ein Modul mit RAM-Bestückung, auf Platz 2 genügt einer für den zweiten EPROM des Betriebssystems. Die Plätze 3 und 4 bleiben für eventuelle Erweiterungen frei. Beim Einfädeln zeigt die Bestükkungsseite des Moduls zum DS 8212 D. Um Fehlverbindungen auszuschließen, sollten ruhig alle 29 Bussignale der Speicherplatine mit Drähten im Kontaktkamm versehen sein, auch wenn einzelne Signale (wie P34) gar nicht benötigt werden. Es folgen das Ausrichten der Anschlußdrähte entsprechend der gewünschten Lage der Speicherplatine (parallel oder senkrecht zur Prozessorplatine), das Einlöten und das Abschneiden der überschüssigen Drahtenden.



17 Bestückungsseite Speichermodul Variante B



18 Bestückungsplan Speichermodul Variante B

Wie nehmen wir den Computer in Betrieb?

Nach gründlicher Sichtkontrolle wiederholen wir den Testbetrieb mit der 5,45-kHz-Schwingung. Kommt er jetzt nicht zustande, liegt eine Fehlverbindung auf dem Speichermodul oder an den Lötstellen vom Systembus vor. Wenn nach eventuell nötigen Korrekturen das gewohnte Schwingen erreicht ist, können trotzdem noch die Adreßbits AØ bis A7 falsch beschaltet sein. Wegen des zwischengeschalteten DS 8212 D führt so ein Fehler nicht zum Abbruch des Testbetriebes. Deshalb überprüfen wir die Signale AØ bis A7 als nächstes auf gleichmäßige Impulsbreite bzw. Lautstärke.

Wenn soweit alles klappt, ist zunächst gesichert, daß die Modulplatine den Prozessor nicht überlastet. Nun können wir die statischen Signale überprüfen. Bei den Modulvarianten B oder C wird eine zu geringe Versorgungsspannung auffallen. Um Batteriestützung (vgl. Abb. 27 bis 30) gestalten zu können, kommen hier nämlich die Sonderspannun-

19 Schaltbild Speichermodul Variante C

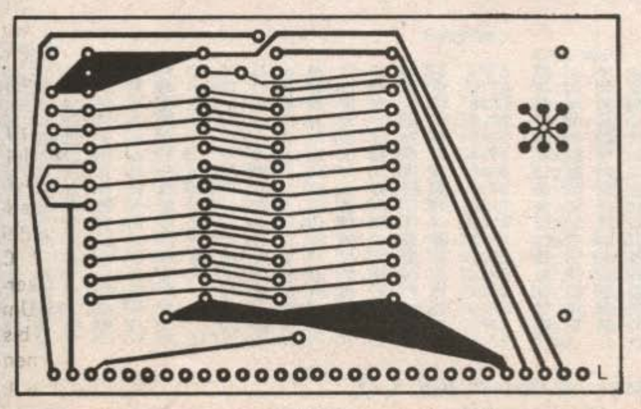
gen U1 und U2 zum Einsatz. Beide müssen auf der Grundplatine z. B. an den Anschlußpunkten neben dem DS 8212 D (natürlich bei abgeschaltetem Rechner) mit 5P verbunden werden, solange keine Batteriestützung eingebaut ist. An den Steuereingängen $\overline{CS}$, $\overline{OE}$ und $\overline{WE}$ muß eine Spannung von ca. 3,5 V anliegen, wenn sich der Computer im Testbetrieb befindet.

An den übrigen Pins der Speicherschaltkreise ist dann die 5,45-kHz-Schwingung zu überprüfen. Bei den Modulvarianten A und D fehlt sie nicht nur an den Masse-, Betriebsspannungs- und Steuereingängen, sondern auch am Pin 21 des U 2716 C. Bei weiteren „Fehlanzeigen" müssen wieder Leiterzugunterbrechungen gesucht werden.

Allgemein ist die Inbetriebnahme am einfachsten, wenn erst Modulplatz 1 und dann anschließend Modulplatz 2 bestückt und getestet werden. Die EPROM brauchen bei dieser Inbetriebnahme noch nicht mit dem Betriebssystem geladen (programmiert) sein, können aber schon stecken.

Wie erhalten die EPROM ihren Inhalt?

Zum Programmieren der beiden EPROM muß man die Hilfe eines Computerklubs oder einer Bildungseinrichtung mit geeignetem Programmiergerät in Anspruch nehmen. Wer keine derartige Adresse kennt, kann sich bei der Redaktion JUGEND+TECHNIK, PF 43, Berlin, 1026 (Tel. Berlin 2 23 34 27), danach erkundigen. **Abb. 26** (Seiten 24/25) zeigt den Speicherinhalt in hexadezimaler Form in Blöcken zu je 256 byte. Die Prüfsummen (CRC) sind für jeden EPROM und für jeden Block angegeben. Wer ein Programmiergerät besitzt und bereit ist, das Betriebssystem ko-



20 Lötseite Speichermodul Variante C

DUMP **CRC 8179** **EPROM 1 (U 2716 C)**

CRC 5D53

```
Ø8ØØ: 3Ø 74 FF 3Ø 76 FF 3Ø 78 FF 3Ø 7A FF 3Ø 7C FF 3Ø
Ø81Ø: 7E FF 8D 2Ø ØØ 8D Ø8 1D E4 15 5A 8B ØA D6 Ø8 24
Ø82Ø: E4 5A 13 AF D6 ØC 1D A6 5A 7F 6B 3F A6 5A 2Ø FB
Ø83Ø: 41 A6 5A 1B 6B 3F E6 5C FØ A6 5A 1A 6B 3A E6 5C
Ø84Ø: 1Ø A6 5A ØA 6B 32 E6 5C FF A6 5A ØB 6B 2A A6 5A
Ø85Ø: Ø8 EB ØD D6 Ø8 78 E6 5A 2Ø D6 ØB 95 E6 5A Ø8 AF
Ø86Ø: A6 5A ØD EB Ø7 D6 ØA D4 E6 5A ØD AF A6 5A ØC 6B
Ø87Ø: 6C AF D6 ØB 95 E6 5C Ø1 Ø4 5C 5B 4B 1C 5B 7D E4
Ø88Ø: 5C 6Ø E4 5B 5C 56 5C ØF A6 5C ØD 7B 4F 9Ø 6Ø FB
Ø89Ø: Ø3 26 5B Ø6 Ø6 5B Ø3 CB 43 7Ø FD 31 6Ø ØC FD 1C
Ø8AØ: ØØ 2C FD 3C 1Ø 4C 7Ø 82 52 92 5Ø 1E 3E 4A F8 4C
Ø8BØ: 1Ø 5C 2Ø 92 5Ø 1E 4A FB 1C 38 2E ØE 3C 7Ø 39 5B
Ø8CØ: 82 52 92 5Ø AØ EØ AØ E2 A6 E3 FØ 7B F3 5C FF A2
Ø8DØ: 25 7B ED 4C 38 92 5Ø 1E 4A FB 5Ø FD AF 7Ø FD 31
Ø8EØ: 6Ø 4C FD 5C 8Ø 3C 2Ø 2C 8Ø ØØ E5 92 34 2A FA 2C
Ø8FØ: Ø2 3C FF 4E 92 34 5A FC 2A F9 5Ø FD BØ 5B AF 31
```

CRC 6A56

```
Ø9ØØ: FØ FC ØØ EC FE BC 1Ø 9C Ø8 8C 92 5C Ø7 4C 4Ø 1C
Ø91Ø: Ø3 E6 Ø6 EØ BØ Ø7 BØ Ø8 BØ F7 D6 Ø8 DD 9F E6 58
Ø92Ø: 14 D6 ØB F2 31 1Ø E6 5A 4B D6 ØB 95 D6 Ø8 24 68
Ø93Ø: 5A FC 16 D6 Ø3 D9 FB 15 26 E6 31 Ø2 66 E3 26 6E
Ø94Ø: E3 36 E6 5A 3D D6 Ø8 72 D6 ØE 92 8B D7 A6 E6 2A
Ø95Ø: EB ØA 76 ØF Ø8 6B 36 56 ØF F7 8B 1C E6 ØF Ø4 BØ
Ø96Ø: ØE Ø8 Ø6 18 Ø7 A6 E6 Ø8 EB Ø6 4C ØØ 92 4Ø 8B B4
Ø97Ø: A6 E6 2B EB 36 D6 Ø6 C9 D6 Ø7 38 D6 ØA D4 E6 58
Ø98Ø: ØC 76 ØF Ø2 EB ØC ØØ 58 76 ØF Ø8 EB Ø5 D6 ØE 8Ø
Ø99Ø: 8B 92 D6 ØB F2 7Ø EØ 7Ø E1 8Ø EØ D6 Ø5 ØØ 28 E4
Ø9AØ: 38 E5 D6 ØE 92 5Ø E1 5Ø EØ 8B E5 A6 E6 2D E6 6E
Ø9BØ: ØC EB 26 E6 6F E6 82 2Ø 42 22 6B D4 AØ EØ 82 3Ø
Ø9CØ: 56 E2 7F D6 ØA A3 D6 ØD CC FB C2 BØ 6D D6 ØC 1D
Ø9DØ: A6 5A 2D EB BB AØ EØ 8B DD D6 Ø3 E7 FB 5A 38 E6
Ø9EØ: D6 Ø2 F4 FC 16 8C 8Ø 42 84 98 E5 42 95 EB Ø1 9E
Ø9FØ: 82 2Ø 42 22 6B 1D 56 E2 7F AØ EØ 82 3Ø D6 Ø1 41
```

CRC C257

```
ØAØØ: 76 ØF 4Ø EB 2E 76 ØF 2Ø EB Ø7 D6 Ø5 93 AØ EØ 8B
ØA1Ø: DF 8Ø EØ E6 6F D7 68 E8 D6 ØC A4 68 E9 D6 ØD CC
ØA2Ø: 7B ØB D6 ØA 81 6E 5B Ø8 D6 ØE 8Ø 8B Ø3 D6 ØC A4
ØA3Ø: 8D Ø9 24 D6 ØA 81 8B DB A6 E6 5F EB Ø7 8F D6 27
ØA4Ø: DF 9F 8B EC A6 E6 4Ø EB E7 19 5Ø Ø9 51 82 6Ø AØ
ØA5Ø: EØ 42 66 EB F8 19 52 Ø9 53 D6 Ø8 DD D6 26 AD 9F
ØA6Ø: 8B CE AØ FE AØ FE 2Ø 5D EB 13 ØC FC 1C ØØ 2C ØC
ØA7Ø: 3C ØD 39 5A D6 Ø8 27 82 3Ø AØ EØ 2A F5 9F 8D Ø8
ØA8Ø: 24 D6 Ø5 ØØ 28 EØ 38 E1 48 EØ 58 E1 AØ E4 AØ E4
ØA9Ø: 82 74 9Ø E7 BB F8 82 74 92 72 AØ E2 AØ E4 42 77
ØAAØ: EB F4 AF D6 Ø1 82 A6 E4 2Ø 6B Ø3 D6 ØA BD BC 15
ØABØ: E3 4B A6 E4 3Ø EB ØB BE A6 EB 1A 7B F3 49 5A 8D
ØACØ: Ø8 72 D6 ØA BD BE E3 4B A6 EB 1A 7B F5 AF E6 5A
ØADØ: 2Ø D6 Ø8 72 76 5B ØF EB F5 AF 7Ø FD 31 5Ø 4A 5E
ØAEØ: 68 FE 78 FF E6 FE FE BØ FF 4C 4Ø FØ 8Ø 46 Ø3 8Ø
ØAFØ: 56 Ø3 7F 76 6D 8Ø 6B Ø5 B6 Ø3 4Ø 8B Ø5 56 Ø3 3F
```

CRC BCD3

```
ØBØØ: ØB FE ØB FE FF 5Ø 8Ø FF 5Ø 8Ø FF 5Ø 8Ø FF 5Ø 8Ø
ØB1Ø: FF 5Ø 8Ø FF 5Ø 8Ø FF 5Ø 8Ø FF 5Ø 8Ø 5E 6B Ø8 26
ØB2Ø: FF Ø8 36 FE ØØ 8B C4 5C FD FF FF 4A BE 56 FA EF
ØB3Ø: 46 Ø3 8Ø 56 Ø3 7F 69 FE 79 FF 4C 78 8B 1C A6 E4
ØB4Ø: 2Ø 7B ØD A6 E4 25 FB ØC 56 Ø3 7F 46 Ø3 8Ø 8B ØA
ØB5Ø: AØ 56 8Ø 56 46 Ø3 8Ø 56 Ø3 7F 5Ø FD BF 18 E2 Ø2
ØB6Ø: 12 Ø2 12 Ø6 E1 EØ ØC ØF AF 58 5B 56 E5 ØF 38 E5
ØB7Ø: 9Ø E5 9Ø E5 Ø2 35 58 E3 56 E3 Ø7 56 E5 38 9Ø E5
ØB8Ø: FØ E5 68 5B FØ E6 4C FE 56 E6 Ø7 6E Ø6 E5 38 16
ØB9Ø: E4 ØØ 6A F8 AF 7Ø FD 31 6Ø ØC FD 18 5B 28 5A 92
ØBAØ: 2Ø D6 ØB 5D 2C Ø3 D6 ØB 69 C2 6Ø D6 ØB BB C2 6Ø
ØBBØ: FØ E6 D6 ØB BB 1E 2A F1 5Ø FD AF 82 74 AØ E4 82
ØBCØ: 84 42 33 6B ØA A8 E3 9Ø E8 1Ø E7 1Ø E9 AA F8 56
ØBDØ: E7 ØF 56 E6 FØ 42 76 42 33 6B ØA A8 E3 EØ E9 CØ
ØBEØ: E7 CØ E8 AA F8 92 84 8Ø E4 92 74 Ø6 E5 Ø8 16 E4
ØBFØ: ØØ AF 7Ø FD 31 5Ø EC ØE FC 98 9C 5A C3 9E 76 EA
```

CRC A3CC

```
ØCØØ: 8Ø 6B F7 ØØ E8 EB F3 56 EA 7F D6 Ø8 27 9C 5A C3
ØC1Ø: 9E 76 EA 8Ø 6B F4 AC 2Ø 5Ø FD 8D Ø8 72 A6 54 75
ØC2Ø: EB FB 7Ø FD 31 6Ø 2C Ø5 D6 ØB 69 6C ØØ D6 ØB BB
ØC3Ø: 2A F9 ØC FD 18 5B 82 2Ø 29 5A 5Ø FD A6 54 75 EB
ØC4Ø: FB D6 ØB 95 D6 ØC 56 76 6D 7F 6B D1 E4 6D 5A 56
ØC5Ø: 6D 8Ø 56 5A 7F AF 7Ø FD 31 6Ø ØC 7F 1C ØF C2 2Ø
ØC6Ø: 56 E2 FØ EB Ø7 1A F7 52 DØ 5Ø FD AF 42 DD EB F9
ØC7Ø: DC 8Ø 3C 4Ø 26 E3 1Ø 9Ø E2 FB F9 42 13 ØC ØF C2
ØC8Ø: 3Ø ØC FF 1C FF 82 2Ø A6 E3 15 6B 12 42 22 EB ØA
ØC9Ø: B6 E3 1Ø A6 E3 1D EB Ø2 3C 7F 42 D3 2C ØØ 6Ø E2
ØCAØ: 92 2Ø 8B C5 28 EØ 38 E1 82 72 AØ E2 42 77 EB F8
ØCBØ: 92 72 8Ø E2 A2 31 6B Ø8 8Ø E2 82 72 AØ E2 8B FØ
ØCCØ: A2 Ø2 7B F4 92 6Ø AØ EØ AF D6 ØC A4 6B Ø8 D6 Ø8
ØCDØ: 24 68 5A A6 E6 6Ø AF D6 ØC C9 6B FA A6 E6 2Ø EB
ØCEØ: F2 D6 ØC CE 8B F6 AØ EØ 82 6Ø 69 5A D6 Ø8 27 42
ØCFØ: 66 EB EØ AF D6 ØD 32 FB 71 D6 ØD Ø9 EB 58 D4 6E
```

CRC 8446

```
ØDØØ: FB 68 D6 ØD Ø9 6B 2B 8B 2D A6 E6 3C 6B 5C A6 E6
ØD1Ø: 3D 6B 57 A6 E6 3E AF A6 6F E6 6B Ø3 E6 6F C9 D4
ØD2Ø: 6E FB 47 A6 E6 22 EB F7 A6 6F E6 6B Ø3 E6 6F D7
ØD3Ø: 3Ø 6E D4 6E FB 34 D6 ØD 58 FB 2F AC Ø3 BC C1 DC
ØD4Ø: Ø3 D6 Ø3 AF 6B EC A6 E6 24 EB 1E D4 6E FB 1B DC
ØD5Ø: Ø3 D6 Ø3 AF 6B DC CF AF A6 E6 28 EB ØE D6 ØD 32
ØD6Ø: FB Ø8 A6 E6 29 EB EF 3Ø 6E DF AF D6 Ø3 D9 FB 1E
ØD7Ø: D4 6E FB F6 D6 Ø3 E2 FB FØ D4 6E FB ED A6 E6 3B
ØD8Ø: 6B E7 A6 E6 ØD 6B E2 A6 E6 5D EB ED 3Ø 6E 4C Ø5
ØD9Ø: D6 Ø3 E7 7B 18 A6 E6 2D 6B 12 A6 E6 25 EB B7 D4
ØDAØ: 6E FB C7 D6 Ø3 F2 FB 1Ø 4A F5 CF AF 4E D4 6E FB
ØDBØ: B9 D6 Ø3 E7 FB Ø2 4A F5 EF AF Ø2 27 ØD 37 23 23
ØDCØ: ØD ØD 27 ØD 23 23 3A 23 6Ø 6Ø 5Ø FF E6 5A 41 D6
ØDDØ: ØB 95 BØ 58 D4 6E FB 13 2C Ø7 3C C8 DC 17 C3 D2
ØDEØ: 2Ø 58 A2 67 6B Ø6 A6 58 12 7B F1 AF 38 58 Ø6 E3
ØDFØ: B9 E6 5C FF D6 Ø8 78 D6 ØB F2 2C ØD 4C ØE C2 52
```

CRC 17A2

```
ØEØØ: 3Ø E4 D4 6E D6 Ø3 D9 FB E2 D4 6E 6C 3D D6 ØD 32
ØE1Ø: FB D9 A6 E6 2C 6B EB A6 E6 3B 6B BØ A6 E6 ØD 6B
ØE2Ø: 14 CF AF D4 6E 8B FØ D4 6E FB CØ A6 E6 3B 6B 9C
ØE3Ø: A6 E6 ØD EB F2 DF AF 5C ØØ ØD 5C Ø1 D6 ØC F4 FB
ØE4Ø: AA 42 55 EB C8 6C 3B E6 58 13 D6 ØB F2 8D ØD CC
ØE5Ø: D4 6E A6 E6 22 EB Ø3 D6 ØD 17 D6 Ø3 D9 7B C4 AF
ØE6Ø: D4 6E A6 E6 22 EB Ø3 D6 ØD 17 A6 E6 3B 6B DE A6
ØE7Ø: E6 ØD 6B C1 D6 ØD 36 FB A8 A6 E6 2C EB 99 8B EØ
ØE8Ø: D6 ØA D4 E6 58 12 D6 ØB F2 BØ E2 38 ØE 56 E3 FØ
ØE9Ø: FØ E3 D6 ØA A3 8D ØA D4 CC 45 54 DØ 52 4F 43 C7
ØEAØ: 4F 54 4F C9 46 C5 4C 53 45 D2 45 54 55 52 4E C7
ØEBØ: 4F 53 55 42 D7 41 49 54 D2 45 4D C3 41 4C 4C D3
ØECØ: 54 4F 5Ø C5 4E 44 D4 52 41 5Ø D4 4F 46 46 DØ 52
ØEDØ: 49 4E 54 DØ 54 48 C9 4E 5Ø 55 54 C5 52 52 4F 52
ØEEØ: 8B 54 48 45 4E C3 4F 4D 5Ø 2Ø 4A 55 2B 54 45 2Ø
ØEFØ: 52 2Ø 5C 5C 5C 5C 5C 5C 5C 5C 5C 5C 5C 5C ØA FF
```

CRC 52BØ

```
ØFØØ: ØØ 2Ø 59 58 43 56 42 4E 4D 2C 2E 2F ØD 1B ØØ ØØ
ØF1Ø: ØØ 15 41 53 44 46 47 48 4A 4B 4C 2A 2D ØA ØØ ØØ
ØF2Ø: ØØ 3D 51 57 45 52 54 5A 55 49 4F 5Ø 2B ØB ØØ ØØ
ØF3Ø: ØØ ØA 31 32 33 34 35 36 37 38 39 3Ø Ø8 1A ØØ ØØ
ØF4Ø: FF FF FF DD DF DF AA FF FF 5Ø 5Ø 5F 69 69 6F 6D
ØF5Ø: B7 6F 6D B7 AF DD FF FF DB BB DF BD DD BF 69 Ø9
ØF6Ø: 6F FD 8D FF FF FD BF FF ØF FF FF FF BF ED B7 7F
ØF7Ø: 94 62 9F D9 DD 8F 1E 97 ØF 1E 9E 1F 75 ØD DF Ø7
ØF8Ø: 1E 1F 97 16 9F ØE DB BF 96 96 9F 96 8E 9F FB FB
ØF9Ø: FF FD FD BF DB 7B DF FØ FØ FF BD ED BF 96 DF DF
ØFAØ: 86 47 8F 96 6Ø 6F 16 16 1F 87 77 8F 16 66 1F Ø7
ØFBØ: 17 ØF Ø7 17 7F 87 46 8F 66 Ø6 6F 8D DD 8F ØE E6
ØFCØ: 9F 65 35 6F 77 77 ØF 6Ø Ø6 6F 62 46 6F 96 66 9F
ØFDØ: 16 17 7F 96 64 8F 16 15 6F 87 9E 1F 8D DD DF 66
ØFEØ: 66 9F AA AA DF 66 ØØ 9F 66 96 6F AA DD DF ØD B7
ØFFØ: ØF 9B BB 9F FØ 66 ØF 9D DD 9F 7B DE EF FF FF ØF
```

26 Hexlisting des Betriebssystems in Blöcken zu je 256 byte (vgl. S. 23)

DUMP CRC 6C9C EPROM 2 (U 2716 C)

CRC 9A6Ø

```
2ØØØ:  8F 31 FØ FC ØØ EC FE BC 1Ø 9C Ø8 8C 92 5C Ø7 4C
2Ø1Ø:  4Ø 1C Ø3 BØ 6D 31 6Ø CC Ø4 E6 7C ØA E6 7D DA C9
2Ø2Ø:  58 D6 21 ØA CA F9 9F D6 Ø8 24 A6 5A 42 6D Ø8 FF
2Ø3Ø:  A6 5A 53 EB Ø3 D6 26 8F A6 5A 4C EB Ø3 D6 27 D1
2Ø4Ø:  A6 5A 52 EB Ø3 D6 23 52 A6 5A 5Ø 6D 23 E4 A6 5A
2Ø5Ø:  49 EB AD D6 Ø8 24 A6 5A 21 EB A5 8F ØC FD 1C ØØ
2Ø6Ø:  2C ØØ 3C Ø8 8Ø EØ 92 2Ø A2 3Ø 7B F8 8B 92 DØ 52
2Ø7Ø:  4F 47 8D 2Ø 2Ø 5Ø C4 41 54 41 8C 2Ø 45 4D 52 2D
2Ø8Ø:  45 53 2Ø 31 39 38 38 ØD 2Ø ØD 2Ø 2Ø 42 2Ø 42 41
2Ø9Ø:  53 49 43 ØD 2Ø 2Ø 49 2Ø 49 4E 49 54 ØD 2Ø 2Ø 53
2ØAØ:  2Ø 53 41 56 45 ØD 2Ø 2Ø 4C 2Ø 4C 4F 41 44 ØD 2Ø
2ØBØ:  2Ø 52 8C 4D 54 42 D6 2Ø B9 FØ 5D E4 5D 5A 56 5A
2ØCØ:  ØF A6 5A ØA 7B Ø3 Ø6 5A Ø7 Ø6 5A 3Ø 8D Ø8 72 7Ø
2ØDØ:  FD 31 5Ø CC 5D 83 CE 5Ø FD 8B DB E4 5E 5D D6 2Ø
2ØEØ:  B6 E4 5F 5D 8B DØ E6 6B Ø3 A6 5D D4 6B 15 E4 5D
2ØFØ:  6A 56 6A ØF A6 6A ØE FB ØD A6 6A ØC 6B Ø5 76 6A
```

CRC D873

```
21ØØ:  Ø4 EB Ø6 ØØ 6B AF E6 6B Ø1 AF 7Ø FD 31 5Ø EC 2Ø
211Ø:  FC 6E 8D ØB FA 26 E1 Ø4 36 EØ ØØ 82 3Ø 8Ø EØ 82
212Ø:  2Ø Ø6 E1 Ø5 16 EØ ØØ AF D6 21 15 22 35 32 24 AF
213Ø:  Ø8 52 18 53 82 2Ø 42 22 6B ØF D6 21 28 42 23 6B
214Ø:  Ø8 26 E1 Ø6 36 EØ ØØ 8B EB CF 82 2Ø 42 22 6B Ø4
215Ø:  A6 E2 3A DF AF Ø8 52 18 53 2C 69 82 3Ø A3 32 EB
216Ø:  1Ø 8Ø EØ ØØ E2 A6 E2 66 FB F1 Ø6 E1 Ø4 16 EØ ØØ
217Ø:  AF 42 33 EB Ø2 DF AF 26 E2 63 22 12 36 EØ ØØ 8B
218Ø:  D8 D6 21 3Ø FB Ø7 EB Ø5 D6 21 41 8B F7 26 E1 Ø5
219Ø:  36 EØ ØØ 2C Ø6 3C 64 93 3Ø 2A FC AF D6 21 55 FB
21AØ:  D4 D6 21 3Ø 6B E7 D6 21 41 8B F9 A8 52 B8 53 Ø8
21BØ:  EA 18 EB D6 21 49 FB E3 6B 3D 2C Ø3 3C 66 9C 3A
21CØ:  8Ø EØ 8Ø EØ 83 3Ø 2A FC D6 21 55 7B CE D6 21 15
21DØ:  Ø8 EA 18 EB 48 E2 58 E3 D6 21 15 82 62 76 E6 Ø4
21EØ:  6B ØC Ø4 71 E5 14 7Ø E4 AØ E2 92 42 8B Ø5 22 53
21FØ:  26 E5 Ø2 AØ E2 92 52 26 EB Ø6 36 EA ØØ 8B BØ 68
```

CRC F5ØD

```
22ØØ:  E4 78 E5 AC FC AØ E6 82 B6 42 BB EB F6 AE EB F5
221Ø:  AF D6 21 FF A8 E6 B8 E7 8Ø EA 88 E6 98 E7 22 95
222Ø:  32 84 82 CA 92 C6 8Ø E6 8Ø EA 8Ø E8 EB F4 AC FF
223Ø:  BC FF 92 A4 Ø8 52 18 53 D6 21 28 7B 1C 26 E1 Ø4
224Ø:  36 EØ ØØ 82 3Ø 22 3B 92 3Ø 8Ø EØ 82 2Ø 32 2A 92
225Ø:  2Ø 8Ø EØ D6 21 49 7B EØ AF 26 E1 Ø6 36 EØ ØØ 8B
226Ø:  F2 D6 21 FF 82 C4 C9 5D D6 2Ø E6 7Ø EB Ø2 B5 BØ
227Ø:  EA 12 A4 22 7B 32 6A 88 E4 98 E5 82 CA 92 C8 AØ
228Ø:  EA AØ E8 8Ø E6 EB F4 D6 21 3Ø FB 22 88 EØ 98 E1
229Ø:  A8 EØ B8 E1 26 EB Ø6 36 EA ØØ 6C Ø6 82 CA 92 C8
22AØ:  8Ø E8 8Ø EA 6A F6 82 C8 42 CC EB EE 8B D9 AC ØØ
22BØ:  5Ø EB 8B 8Ø D6 Ø8 24 A6 5A 3Ø 7B Ø5 A6 5A 46 3B
22CØ:  Ø2 DF AF A6 5A 3A 6B F9 CF AF 31 5Ø BC 7Ø E8 7Ø
22DØ:  F8 71 D6 2Ø DB BE 31 6Ø CC Ø6 E6 5A 2Ø D6 Ø8 72
22EØ:  CA FB 26 5B Ø6 D6 22 B4 7B Ø9 C8 5A D6 22 B4 FB
22FØ:  Ø3 42 CC AF D6 23 Ø6 38 5A C9 5A D6 23 Ø6 C8 5A
```

CRC 4EF9

```
23ØØ:  FØ EC 42 C3 CF AF A6 5A 3A 7B Ø3 Ø6 5A Ø9 56 5A
231Ø:  ØF AF A6 5A 2B EB Ø3 AØ EE AF A6 5A 47 EB FA D6
232Ø:  22 E5 FB 24 EB F3 CC Ø3 D6 Ø8 24 7Ø 5A CA F9 CC
233Ø:  Ø3 9C 68 51 E9 ØØ E9 CA FA 9C 3A D6 21 55 7B D9
234Ø:  D6 21 15 E8 E2 F8 E3 AF E8 EC D6 22 E5 7B CA F8
235Ø:  EC AF D6 Ø8 DD 7Ø FD 31 6Ø E9 7Ø F9 71 31 5Ø 8C
236Ø:  Ø3 D6 21 ØA E8 6E F8 6F D6 2Ø DB D6 ØA CE 31 6Ø
237Ø:  CC Ø6 BC Ø4 E4 5F 5A D6 2Ø BE E6 5A 2Ø D6 Ø8 72
238Ø:  42 EE 6B Ø5 D6 2Ø CF 8B Ø8 E5 5F 5D 2Ø 5F D6 2Ø
239Ø:  B6 BA E7 CA DD D6 22 CA 7B 15 31 7Ø 42 ØØ 6B Ø6
23AØ:  2C 6C 93 2Ø 8B Ø5 F5 6C E1 AØ EØ BØ 5B 8B AE A6
23BØ:  5A 6Ø 7B Ø3 5Ø FD AF A6 5A ØD EB Ø9 DC ØØ Ø6 EF
23CØ:  Ø4 12 ED 8B 1A A6 5A 2D EB 12 8Ø 7Ø 48 7Ø 58 71
23DØ:  22 5F 32 4E FB D5 E8 7Ø F8 71 8B CF D6 23 12 BØ
23EØ:  5B 8D 23 57 7Ø FC 7Ø FD 31 5Ø 8C Ø5 D6 21 ØA E8
23FØ:  E2 F8 E3 D6 2Ø DB BC Ø4 31 6Ø D6 23 1F 7B Ø4 E9
```

CRC 7EF3

```
24ØØ:  52 F9 53 5Ø FD 8F D6 Ø8 DD E6 58 Ø1 D6 21 ØA 5Ø
241Ø:  6C 7Ø 6C 7Ø FD 31 5Ø FC Ø8 AC 3Ø 9Ø 6C 12 A8 D6
242Ø:  Ø8 72 FA F5 F8 6F E8 6E BØ 7Ø 82 DE D6 2Ø E6 F9
243Ø:  71 Ø4 6B 71 14 EE 7Ø D6 26 ØE A6 5B 7Ø 7B F8 9F
244Ø:  D6 22 CA FB 2C ED 24 D3 E6 5B 73 D6 ØB 95 E6 5B
245Ø:  7Ø CC Ø3 D6 Ø8 24 7Ø 5A CA F9 CC Ø3 9C 68 51 E9
246Ø:  ØØ E9 CA FA 9C 3A 48 7Ø 58 71 D6 21 9C FB 15 8B
247Ø:  CF 31 7Ø 38 6C 39 5D 92 3Ø 31 6Ø D6 2Ø E6 E8 7Ø
248Ø:  F8 71 BA Ø5 5Ø FD 8D 24 Ø5 D6 22 E5 FB 32 EB F4
249Ø:  A6 73 D6 6B ØD 56 73 ØF A6 73 ØC 6B E7 A6 73 ØA
24AØ:  7B E2 CC Ø3 D6 Ø8 24 7Ø 5A CA F9 CC Ø3 6C 69 51
24BØ:  E6 ØØ E6 CA FA 48 7Ø 58 71 6C 3A D6 21 81 8B C4
24CØ:  AØ 7Ø 31 7Ø 38 6C 92 3Ø ØØ 6B 6B B8 31 6Ø D6 22
24DØ:  E5 8B ED A6 5A 51 EB Ø7 E9 7Ø F9 71 8D 24 4Ø A6
24EØ:  5A 2D EB Ø6 8Ø 6E BØ 6D 8B 9A A6 5A 54 EB 44 7Ø
24FØ:  EE 7Ø EF D6 23 1F BØ 7Ø BØ 71 5Ø E5 5Ø E4 7B Ø8
```

CRC 3DDD

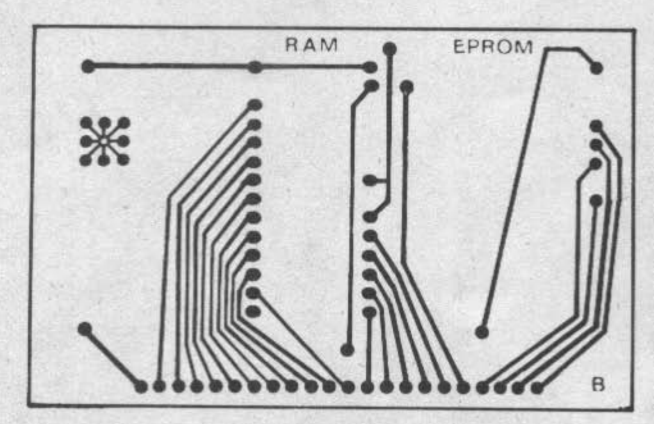
```
25ØØ:  22 F5 32 E4 E9 7Ø F9 71 E8 E4 F8 E5 8F D6 21 AB
251Ø:  FD 24 84 9F 9C 3F 7Ø E9 CC Ø3 9C 68 71 E9 ØØ E9
252Ø:  CA FA E6 5B 75 CC Ø4 5Ø 5A D6 Ø8 72 CA F9 D6 ØA
253Ø:  D4 8B A9 A6 5A 49 48 7Ø 58 71 EB Ø5 D6 22 11 8B
254Ø:  A7 A6 5A 58 EB Ø5 D6 22 61 8B 9D 82 BE B9 5D D6
255Ø:  2Ø E6 A6 5A ØD EB Ø8 BØ ED Ø2 FB 12 ED 8B 89 A6
256Ø:  5A 4E EB ØC BØ 7Ø B9 71 Ø4 EF 71 14 EE 7Ø 8B 17
257Ø:  A6 5A 48 EB 16 7Ø 6E 7Ø 6F D6 23 1F E9 7Ø F9 71
258Ø:  5Ø 6F 5Ø 6E 7D 24 E8 5Ø FD 8B 3C 5Ø FD A6 5A 4C
259Ø:  EB Ø4 5Ø FC 3Ø 6E A6 5A 52 EB 13 7Ø 6E 7Ø 6F BØ
25AØ:  6E E4 FD 6F D6 23 52 5Ø 6F 5Ø 6E 8D 24 Ø5 A6 5A
25BØ:  6Ø 7B Ø3 8D 2Ø ØØ A6 5A 53 6B ØA 7Ø FD 31 6Ø D6
25CØ:  23 12 8D 24 84 BØ 7Ø 8F 56 F1 FC E6 7C 25 E6 7D
25DØ:  E3 56 FA 2F E6 F4 Ø4 5Ø FC 7Ø 6F 7Ø 6E 7Ø FC 46
25EØ:  F1 Ø3 BF 56 F1 FC 5Ø FC 5Ø 6E 5Ø 6F 7Ø FC 56 FA
25FØ:  2F 44 7Ø 7Ø 6B ØA A4 6E 7Ø EB DC A4 6F 71 EB D7
```

CRC 62CC

```
26ØØ:  E6 7C ØA E6 7D DA E6 F4 4Ø 46 F1 Ø3 8B 9D 31 6Ø
261Ø:  48 5E 58 5F D6 21 3Ø FB 1C 6B Ø5 D6 21 41 8B F7
262Ø:  CC Ø4 82 2Ø 7Ø E2 8Ø EØ CA F8 CC Ø4 5Ø 5A D6 Ø8
263Ø:  72 CA F9 8B Ø3 D6 2Ø DB 31 5Ø AC 2Ø D6 Ø8 72 82
264Ø:  DE D6 2Ø B6 D6 2Ø E6 AC 2Ø D6 Ø8 72 31 6Ø 48 5E
265Ø:  58 5F D6 21 3Ø FB 22 EB Ø5 D6 21 41 8B F7 Ø4 6B
266Ø:  5F 16 5E ØØ CC Ø4 82 2Ø 7Ø E2 8Ø EØ CA F8 CC Ø4
267Ø:  5Ø 5A D6 Ø8 72 CA F9 8B 13 AØ 5E ØØ EB 6B ØD D6
268Ø:  2Ø CF E6 5A 2Ø D6 Ø8 72 BA F5 ØØ 5B 8D ØA D4 D6
269Ø:  Ø8 DD D6 23 1F 7B F8 F9 5Ø E9 51 2Ø 5B D6 23 1F
26AØ:  7B Ø4 E9 52 F9 53 E8 52 E4 53 52 E9 53 E6 5B 2Ø
26BØ:  D6 Ø8 24 A6 5A ØD EB F8 8F ØC FC Ø9 Ø3 1C ØØ 2C
26CØ:  FD 3C 2Ø 4C ØB 82 52 92 5Ø 1E 3E 4A F8 92 4Ø 1E
26DØ:  DB FB 5C Ø2 1C 1Ø 92 5Ø 1E 5C 5Ø 4C Ø4 93 5Ø 4A
26EØ:  FC 1C ØØ 6C Ø1 CC 2Ø D6 27 4E Ø8 51 18 5Ø 28 53
26FØ:  38 52 6E 48 E2 58 E3 22 51 32 4Ø FB Ø1 AF 26 E5
```

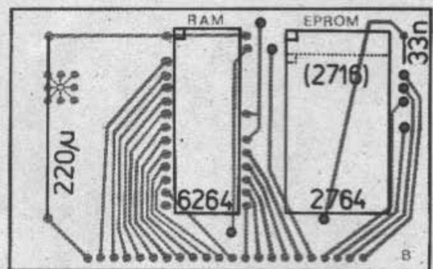
CRC D741

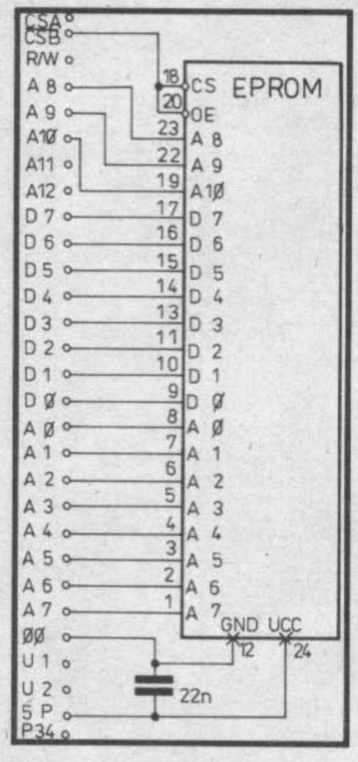
```
27ØØ:  8Ø 32 4C FB Ø2 6C FF DC AØ D6 27 4E 8B E4 4C ØØ
271Ø:  58 E8 8Ø E4 EB FC B6 Ø3 4Ø 58 E8 8Ø E4 EB FC B6
272Ø:  Ø3 4Ø AF BC FF CC FA BE 88 Ø3 56 E8 Ø1 A2 84 6B
273Ø:  F4 CE EB F3 48 E8 AF D6 27 23 D6 27 23 AC Ø8 D6
274Ø:  27 23 D6 27 23 A6 EB 16 EF CØ E9 AA F2 AF 8C 4A
275Ø:  D6 27 ØE 8Ø EC EB F7 98 E6 D6 27 6D DC 8Ø 7C ØØ
276Ø:  82 9Ø AØ EØ D6 27 6D Ø2 79 DA F5 98 E7 8C 94 D6
277Ø:  27 ØE AC Ø8 EØ E9 8C 4A 7B Ø2 8C 25 D6 27 ØE AA
278Ø:  F3 AF DC 5Ø EC 16 FC 2A D6 27 C7 7B F5 DA F9 EC
279Ø:  2A FC 6A D6 27 C7 7B FB D6 27 C7 7B E5 D6 27 3D
27AØ:  99 5D DC 8Ø 7C ØØ 6E D6 27 37 92 9Ø AØ EØ Ø2 79
27BØ:  DA F5 D6 27 37 E6 5B 7Ø D6 2Ø B6 A2 79 EB Ø5 A4
27CØ:  5D E6 6B ØC 8D ØA 62 D6 27 23 A2 BE 7B Ø2 A2 FB
27DØ:  AF BØ Ø6 ØØ 5B D6 23 1F 8F 7B Ø4 E9 Ø6 F9 Ø7 31
27EØ:  4Ø ØC FC 1C ØØ 6C ØØ D6 27 82 2C FC 3C 11 Ø8 Ø6
27FØ:  18 Ø7 42 ØØ EB Ø5 82 12 3E 82 Ø2 D6 27 82 8B FB
```



21 *Bestückungsseite Speichermodul Variante C*

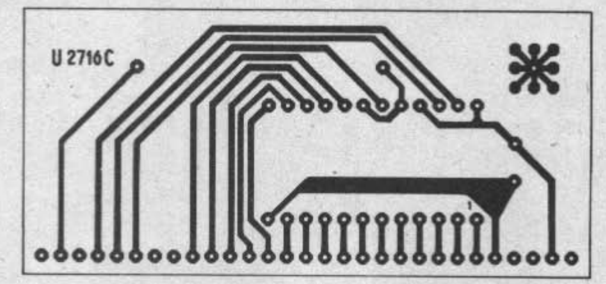
22 *Bestückungsplan Speichermodul Variante C*

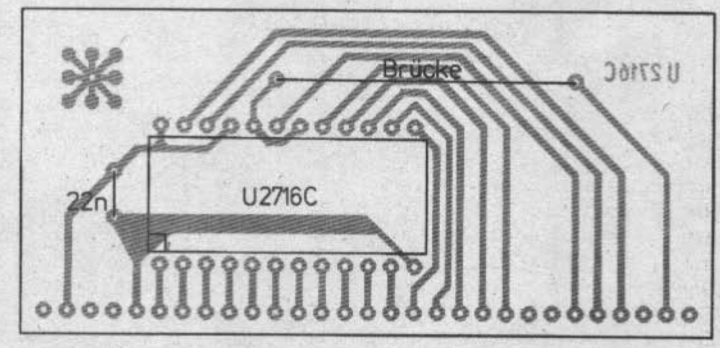




23 *Schaltbild Speichermodul Variante D*

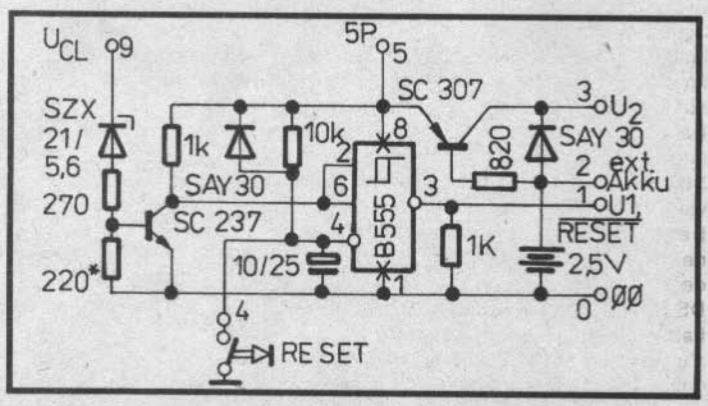
24 *Leiterzugbild Speichermodul Variante D*



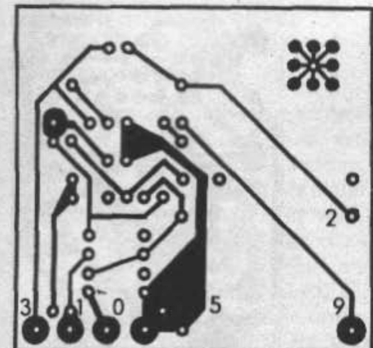


25 *Bestückungsplan Speichermodul Variante D*

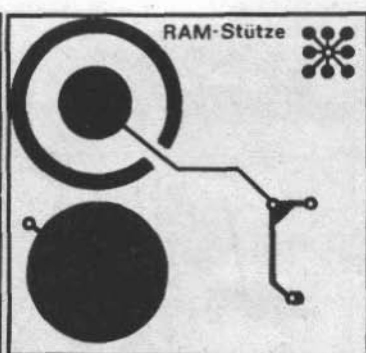
27 *Schaltbild der Batteriestütze für CMOS-RAM-Schaltkreise*



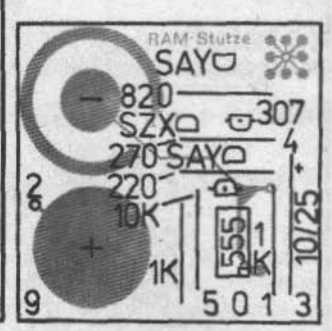
stenlos weiterzugeben, kann über die Redaktion auch das Programmieren von „Mutterschaltkreisen" vermittelt bekommen. Außerdem enthält diese Bauanleitung einen einfachen EPROM-Programmierzusatz. Mit ihm ist man dann selbst in der Lage, das Betriebssystem in weitere EPROM zu übertragen.



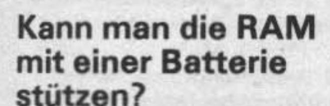
28 Lötseite der RAM-Stütze



29 Bestückungsseite der RAM-Stütze



30 Bestückungsplan der RAM-Stütze (SC 237 nicht bezeichnet)



Kann man die RAM mit einer Batterie stützen?

Auf den Speichermodulvarianten B und C ist das Ergänzen einer Stützbatterie bereits vorgesehen. Sie sichert den Speicherinhalt der U 6516 D oder U 6264 D auch bei abgeschaltetem Computer. So bleiben Programme beliebig lange erhalten, ohne daß sie jedes Mal von Kassette geladen werden müssen. Dazu bekommen die RAM-Schaltkreise ihre Betriebsspannung (U2) bei abgeschaltetem Computer aus Akkumulatoren. Außerdem muß unkontrolliertes Beschreiben der RAM-Zellen, was gewöhnlich beim Ein- und Ausschalten auftreten kann, vermieden werden. Die Schaltung **(Abb. 27)** erzeugt dazu ein $\overline{\text{RESET}}$-Signal für den Einchip-Mikrorechner, wenn die Spannung am Ladekondensator zu klein ist, die Betriebsspannung einen Kurzschluß hat oder die Rücksetztaste betätigt ist. Dieses Signal wird beim Einschalten lange genug aktiv (Ø) gehalten, so daß ein regulärer Programmstart stattfindet.

Das Rücksetzsignal erhalten auch die Speichermodule als Sonderspannung U1 zum Passivieren der Schaltkreisfreigabe. Der U 6264 D besitzt hierfür einen 1-aktiven CS-Eingang. Für den U 6516 D muß der Dekoder DS 8205 D benutzt werden. Deshalb funktioniert die Batteriestützung mit der Modulvariante B nur dann, wenn mindestens zwei RAM-Schaltkreise bestückt sind. Sonst fehlt nämlich der für den Schutz wichtige DS 8205 D (besser: U 74 HCT 138 DC).

Die Leiterplatte der Batteriestützung **(Abb. 28, 29 und 30)** wurde so gestaltet, daß sie direkt an die vier Versorgungsspannungs-Anschlüsse neben dem DS 8212 D auf der Prozessorplatine angeschlossen und dort senkrecht aufgesetzt werden kann. Der Elko am Pin 6 des Prozessors ist zu entfernen. Statt dessen erhält dieser Anschluß U1 über einen Schaltdraht als $\overline{\text{RESET}}$. Die Rücksetztaste wird jetzt am Anschluß 4 der Stützplatine kontaktiert. Ein weiterer Draht verbindet den Anschluß 9 dieser Platine mit dem Pluspol des Ladekondensators (2200 µF/10 V) oder mit Pin 3 des Spannungsreglers B 3170 V.

Unter Umständen muß der 220-Ω-Widerstand der Stützschaltung variiert werden, um das Erzeugen des Nullpegels an U1 bei einer Ladekondensatorspannung von 7,5 V bis 8 V zu erreichen. Als Akkus können wir z. B. zwei „Kosmos"-Zellen direkt befestigen, indem wir auf die Stützplatine Metallstreifen und eine Bronzefeder löten. Sonst dienen zwei in Reihe geschaltete NC- oder Bleizellen als Stützbatterie am Anschluß 2 der kleinen Platine. Der 820-Ω-Widerstand dient dem Nachladen und muß für Blei-Akkus auf 120 Ω verringert werden.

Der Computer funktioniert auch ohne Batteriestützung der CMOS-RAM-Schaltkreise. Dann müssen aber die Sonderspannungen U1 und U2 mit 5P verbunden sein.

4. Ein-/Ausgabeplatine

Wie wird die Standard-Peripherie angeschlossen?

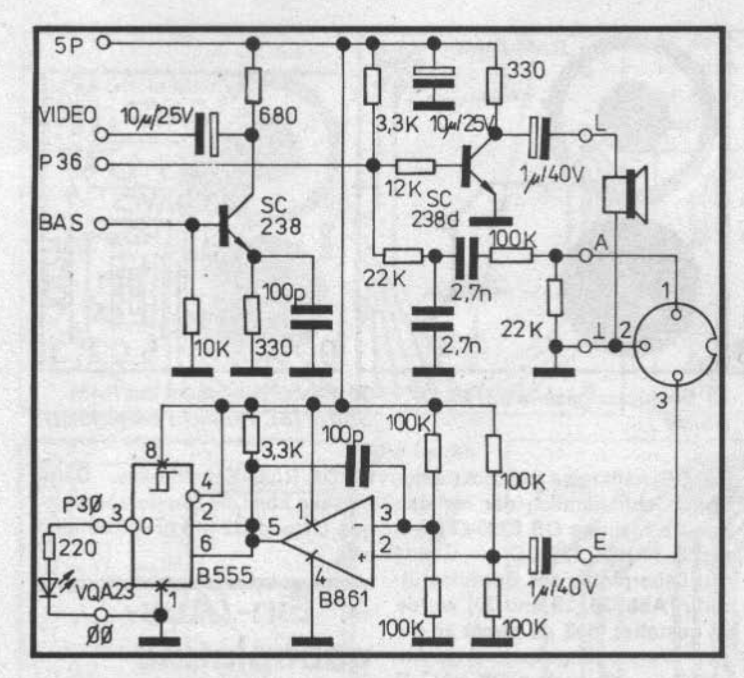
Für den Anschluß der außerhalb des Grundgerätes befindlichen Standardperipherie Bildschirm und Kassettenrecorder benötigt unser Computer eine Ein-/Ausgabe-Schaltung (Interface). Die Schaltung zeigt **Abb. 31**, **Abb. 32** die Lötseite der Platine, **Abb. 33** den Bestückungsplan.

Zur Standard-Peripherie gehört auch die Tastatur, die unter Punkt 7. abgehandelt wird.

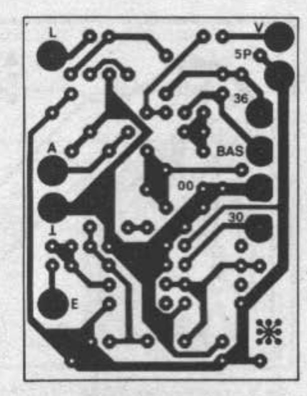
Auf der Prozessorplatine befindet sich der Eingabetreiber der Tastatur. Der Dekoder zum Erzeugen der Tastenauswahlsignale wird am besten in die Tastatur integriert, um die Anzahl der Verbindungsdrähte klein zu halten.

Wie funktioniert die Bildausgabe?

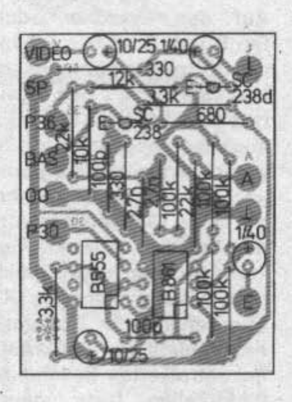
Für die Bildausgabe eignet sich ein Fernsehempfänger. Auf der Prozessorplatte wird aus dem Schieberegisterausgang (Helligkeit) und dem Signal P 37 (Syn-



31 Schaltbild der Ein-/Ausgabeplatine



32 Leiterzugbild der Ein-/Ausgabeplatine



33 Bestückungsplan der Ein-/Ausgabeplatine

chronimpulse) das Bildamplituden- und Synchronsignal BAS gemischt (vgl. Abb. 6). Es enthält positiv gerichtete Impulse und ist so bemessen, daß es sich direkt an den Anschluß 10 des ZF- und Demodulatorschaltkreises A 240 D, also parallel zum Stellwiderstand des Weißabgleichs, anschließen läßt. Beim Herstellen der betreffenden Verbindung zwischen Computer und Fernseher (zum Beispiel Robotron RF 3301) mit abgeschirmtem HF-Kabel sollte man **die Hilfe eines Fachmanns in Anspruch nehmen.**

Verschiedene Fernseher sind mit einem Video-Eingang ausgerüstet. Hier wird das Bildamplituden- und Synchronsignal aber mit negativ gerichteten Impulsen erwartet. Die dafür erforderliche Umkehrstufe ist in der Schaltung der Ein-/Ausgabe-Platine (Abb. 31) enthalten.

Auch in Fernsehern ohne Video-Eingang und ohne A 240 D als ZF-Verstärker läßt sich gewöhnlich ein Punkt finden, an dem das mit der Videostufe erzeugte Signal eingekoppelt werden kann. Beim Junost 402 B bietet sich der Anschluß KT 8 an der Anode der Bilddemodulatordiode an. Wird der Ausgangselko der Videostufe durch eine Brücke ersetzt, bewirkt der Gleichspannungsanteil sogar ein Abschalten des Empfangsteils über die Verstärkungsregelung des Fernsehers. Optimale Verhältnisse ergeben sich, wenn zwischen dem Videosignal und dem Anschluß KT 8 im Fernseher ein Widerstand von 4,7 kΩ geschaltet wird. Für die Masseverbindung eignet sich das Schirmblech in der Nähe dieses Anschlußpunkts gut.

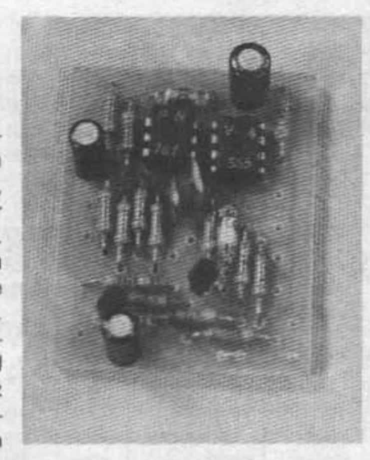
Nicht immer finden sich so günstige Umstände wie beim Junost. Das Vermeiden von Störungen über den normalen Empfangsweg bedarf dann gesonderter Maßnahmen. Moderne Geräte besitzen Schaltspannungseingänge für das Umschalten auf Videobetrieb. Sie betreffen das Abschalten des HF-Teils und das Einstellen eines günstigen Einschwingverhaltens der Bildablenkung. Ohne diese Möglichkeit zu nutzen, lassen sich diese Geräte kaum einsetzen. Für das Ergänzen eines Video-Einganges ist hier deshalb die Hilfe eines Fachmanns unumgänglich.

Keinesfalls darf ein älteres Gerät mit direkter Stromversorgung aus dem Netz verwendet werden, da das mit erheblicher Unfallgefahr verbunden wäre! Ein Ines oder Stella kommt daher nicht in Frage.

Geräte mit Trenntrafo sind aber auch vorsichtig zu behandeln, da sich das Chassis (Masse) trotzdem so stark aufladen kann, daß es den Computer gefährdet. Daher dürfen Schaltarbeiten keinesfalls bei angeschlossenem Fernseher ausgeführt werden. Die Verbindung zum Fernseher ist nur dann herzustellen, oder zu trennen, wenn beide Geräte ausgeschaltet sind.

Ist das Magnetband-interface auch so kompliziert?

Zum Kopieren von Speicherinhalten auf Magnetband muß ein Verfahren angewendet werden, das den Eigenschaften normaler Kassettenrecorder gerecht wird. Unser Computer verwendet dazu die gleiche Methode wie die Kleincomputer KC 85 und KC 87. Es werden drei Frequenzen unterschieden: 600 Hz, 1200 Hz und 2400 Hz. Eine Periode von 600 Hz trennt zwei Bytes. 1200 Hz entspricht Ø-Pegel, 2400 Hz dagegen 1-Pegel. Für jedes Bit wird nur eine Periode gespeichert, also für die Bits eines Bytes acht mit 1200 Hz oder 2400 Hz und eine mit 600 Hz als Trennzeichen. 128 bytes bilden einen Block, den 1 byte Kontrollsumme ergänzt. Den Blockanfang kennzeichnen viele 2400-Hz-Perioden hintereinander (Kennton) und die Blocknummer. Vor dem ersten Block wird ein besonders langer Kennton erzeugt, im ersten Block stehen nur der Name der Datei, die Anfangsadresse und die Endadresse. Das Erzeugen und Dekodieren dieser Signale erledigt unser Computer programmtechnisch. Sogar das Unterdrücken hochfrequenter Störungen realisiert die Software. Die Elektronik der Interface-Platine (Abb. 31) dient eigentlich nur der Anpassung an die Pegel des Recorders. Der Einchip-Mikrorechner gibt Töne über P36 aus. Ein Verstärker mit einem SC 237 gestattet dazu den Anschluß eines Lautsprechers oder besser einer Telefon-Hörkapsel. Für das Aufzeichnen von Programmen mit dem Kassettenrecorder wird der Frequenzgang auf den genutzten Bereich beschnitten. Arbeiten wir mit Handaussteuerung, kann der gegen Masse geschaltete Kondensator von 2,7 nF auf 1 nF verringert werden, um Schwächen des Recorders bei der Höhenwiedergabe auszugleichen. Bei automatischer Pegelsteuerung bewirken zu hohe Frequenzanteile unnötig leise Aufnahmen.



Bestückte Ein-/Ausgabeplatine

Für die Eingabe vom Magnetbandgerät in den Rechner ist ein Verstärker erforderlich. Um keine zusätzliche Betriebsspannung zu benötigen, wird der B 761 D bzw. dessen Anfalltyp B 861 D genutzt. Er erzeugt aus dem Eingangssignal entsprechende Rechteckschwingungen. Der B 555 D dient als Verstärker, der auch eine Kontroll-LED mitbedienen kann. Manchmal neigt der Operationsverstärker trotz der schwachen Rückführung zu Schwingungen, die sich aber mit einem zusätzlichen Kondensator (ca. 20 pF) zwischen dessen Pins 5 und 6 vermeiden lassen.

Die vier Eingangswiderstände bestimmen die Empfindlichkeit. Bei 10 Prozent Abweichung zwischen den Vorspannungen der Operationsverstärker-Eingänge muß das Signal vom Kassettenrecorder mindestens 500 mV betragen, was höchst selten gesichert ist. Entweder es werden 1-%-Widerstände oder ausgemessene Exemplare eingesetzt. Wichtig ist die Gleichheit, der Absolutwert darf zwischen 10 kΩ und 200 kΩ liegen.

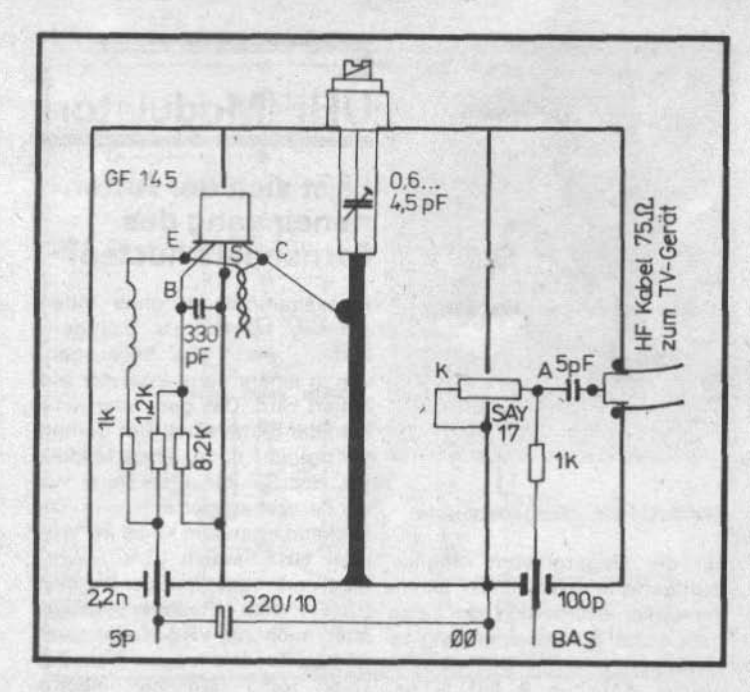
5. UHF-Modulator

Läßt sich der Antenneneingang des Fernsehers nutzen?

Fernsehempfänger ohne Videoeingang können als Sichtgerät dienen, wenn die Bildausgabe wie in einem Fernsehsender moduliert wird. Das geht aber zu lasten der Bildqualität. Der Computer braucht dafür einen Modulator, der für den Empfänger wie ein Fernsehsender erscheint. Dabei kann irgendein Kanal im VHF- oder UHF-Bereich genutzt werden. Verbreitet dient der Kanal 36 (UHF) dem Rechneranschluß, aber auch für VHF-Modulatoren gibt es Bauanleitungen (zum Beispiel: radio fernsehen electronic 1/1985, Seite 13 ff).

Der Eigenbau verlangt handwerkliches Geschick. Nur bei sehr exakter Ausführung kann das Abstrahlen von hochfrequenten Signalen, die den Funkverkehr stören, vermieden werden. Der Aufbau und der Betrieb solcher Modulatoren ist durch die Deutsche Post genehmigungspflichtig. Daher empfiehlt sich der Selbstbau dieses Ergänzungsmoduls nur für Funkamateure, die mit diesen Fragen vertraut sind. Das Ausstatten eines Fernsehempfängers mit Videoeingang macht allgemein viel weniger Schwierigkeiten.

Kern des UHF-Generators ist ein Germaniumtransistor mit sehr hoher Grenzfrequenz (zum Beispiel GF 145, GF 147, AF 139, AF 239). Wie die meisten Bauelemente kann er aus einem ausgedienten UHF-Konverter oder -Tuner ausgelötet werden. Zu seiner Beschaltung gehören zwei Kammern des in **Abb. 34** dargestellten Gehäuses. Es besitzt eine Grundfläche von 37 mm × 40 mm und hat eine Höhe von 10 mm. Es kann am besten aus dünnem Messingblech (≦ 1 mm) gefertigt werden. Der Aufbau verlangt



34 Schaltung und Bauelementeanordnung des UHF-Modulators

hohe Präzision. Die Bohrungen für die beiden Durchführungskondensatoren, den Rohrtrimmer und das HF-Kabel müssen knapp bemessen werden. Alle Lötnähte, die nicht hundertprozentig dicht sind, machen den Modulator wegen Abstrahlung von UHF-Signalen unbrauchbar.

Die erste Kammer enthält die Bauelemente für die Festlegung des Arbeitspunktes des Transistors. Grundsätzlich sind möglichst kleine Bauformen zu verwenden, damit störende Nebeneffekte (parasitäre Induktivitäten und Kapazitäten) begrenzt bleiben. Die Luftspule am Emitter hat zehn Windungen (CuL 0,5 mm) und einen Durchmesser von 5 mm.

Die erste Trennwand besitzt einen Ausschnitt von etwa 6 mm × 8 mm, in den der Transisor eingesetzt wird. Die zweite Kammer (Breite: 14 mm) bildet den Schwingkreis. Wichtig ist, eine hohe Güte zu erzielen. Dazu muß die Induktivität aus versilbertem Draht mit mindestens 1,5 mm Durchmesser bestehen. Die Kollektoranzapfung befindet sich in der Mitte der Kammer. Als Rückführungskapazität dient verdrillter Kupferlackdraht (CuL 0,5 mm), dessen Länge von ca. 10 mm auf sicheres Anschwingen und minimale Dämpfung des Schwingkreises abgeglichen wird.

Die zweite Trennwand besitzt ein Bohrloch für die Modulatordiode (SAY 17). Die Schlaufe deren Katodenanschlusses in der Schwingkreiskammer ist möglichst klein (3 mm höchstens) auszulegen, damit nur soviel Energie entnommen wird, wie der Tuner des Fernsehempfängers benötigt. Das sichert eine minimale Dämpfung des Schwingkreises mit entsprechender Frequenzreinheit und reduziert die Gefahr der Abstrahlung über den Ausgang des Modulators.

Das HF-Kabel mit dem Antennenstecker für den Fernseher-Eingang muß sehr direkten Massekontakt erhalten. Dazu wird die Abschirmung des durchgesteckten Kabels am besten fächerartig abgespreizt und ringsum von innen ausgelötet.

Für den Modulator bleibt genügend Raum auf der Bestückungsseite der Prozessorplatine links oben (vgl. Abb. III. Umschlagseite). Massekontakt besteht durch die Kupferfläche. Die Versorgungsspannung 5P und das Bildamplituden- und Synchronsignal BAS werden gleich rechts über Lötaugen der Lötseite angeboten. Vor dem Einschalten darf das Aufsetzen eines ideal dicht schließenden Deckels nicht vergessen werden.

Ob der Oszillator schwingt, läßt sich indirekt am Stromverbrauch erkennen. Er beträgt normalerweise etwa 10 mA, ohne Schwingen nur etwa 1 mA (Richtwerte). Setzt der Oszillator auch bei vergrößerter Rückkopplung (verdrillter Kupferflachdraht) nicht ein, hat der Transistor meist eine zu kleine Verstärkung und muß ausgetauscht werden. Zum Frequenzabgleich stellen wir am Fernseher den Kanal 36 ein und suchen mit dem Trimmkondensator des angeschlossenen Modulators die Stellung, bei der das Bild am dunkelsten ist und das Rauschen im Lautsprecher verschwindet. Dabei kann durch ständiges Festhalten der $\overline{\text{RESET}}$-Taste die Modulatordiode geöffnet werden.

Wie stellen wir das beste Bild ein?

Nachdem die Prozessorplatine mit aufgesetzten Speichermodulen (Plätze 1 und 2) und gesteckten programmierten EPROM den Testbetrieb wie beschrieben fehlerfrei bis ins letzte absolviert hat, können wir uns mit der Bilderzeugung beschäftigen. Dazu sind Fernseher und Computer miteinander zu verbinden und anschließend einzuschalten. Zuvor trennen wir aber die Verbindung von P32 mit P35 und verbinden P32 mit Masse (00). Nach Betätigen der $\overline{\text{RESET}}$-Taste beginnt der Computer mit dem Erzeugen des Eingangsmenüs, das unter der Überschrift „EMR-ES 1988" sechs Dienste anbietet.

Meist ist das Bild nicht auf An-

hieb lesbar. Mit dem Trimmkondensator 10/40 kann die Phasenlage des Schiebetaktes so eingestellt werden, daß überhaupt helle und dunkle Bildanteile entstehen. Der Einstellregler ausgangs des Schieberegisters bestimmt den Kontrast. Bei zu großem Widerstandswert entsteht ein flaues Bild, bei zu kleinem ein instabiles. Beide Einstellungen bringen wir auf die Mitte des akzeptablen Bereichs und schätzen (besser: messen) den eingestellten Wert. Diesen realisieren wir anschließend durch Widerstand und Kondensator geeigneter Größe als Festwert.

Noch einige Hinweise zur Fehlersuche: Die Synchronimpulse gibt der Einchip-Mikrorechner über P37 aus. Sie bestehen aus einer 15-kHz- (Zeilen) und einer 100-Hz-Schwingung. Sie muß sich auch im BAS- und Videosignal mit Oszilloskop oder Kopfhörer finden lassen. Tritt diese Schwingung nicht ein, läuft das Betriebssystem nicht. Ursache dafür sind fehlerhafte Verbindungen zwischen Einchip-Mikrorechner und den Speicherschaltkreisen, sehr selten auch defekte EPROM oder RAM. Mit dem Testbetrieb lassen sich Verbindungsfehler gut ausschließen, manchmal bleibt jedoch ein fehlendes Steuersignal ($\overline{CS}$, R/W) unerkannt.

Den Bildinhalt steuert das Schieberegister bei. Voraussetzung sind die Übernahmeimpulse am MC-Eingang (8 je Zeile). Sie werden vom DL 030 D erzeugt, der DL 000 D ist ebenfalls beteiligt. Wegen der hohen Frequenz (250 kHz) können wir sie mit dem Oszilloskop, nicht aber mit dem Kopfhörer verfolgen. Senkrechte Streifen auf dem Bild deuten auf korrekte Takterzeugung, aber defekte Schieberegister hin. In jeder Bildzeile kommen die Bildpunkte in acht Zyklen mit je acht Bit zur Anzeige. Links steht jeweils Bit 7, es folgen Bit 6 bis Bit Ø. Im Fehlerfall können wir abzählen, wieviel Flipflops des Schieberegisters funktionieren. Sind weniger als vier in Ordnung, ist der dem V 40098 D nähere D 195 D, sonst der andere nicht funktionstüchtig. Bei genau vier kann die Verbindung zwischen beiden gestört sein. Die Erfahrung besagt, daß sich Ausfälle stark auf ältere Bastlertypen des D 195 D konzentrieren.

6. Tastatur

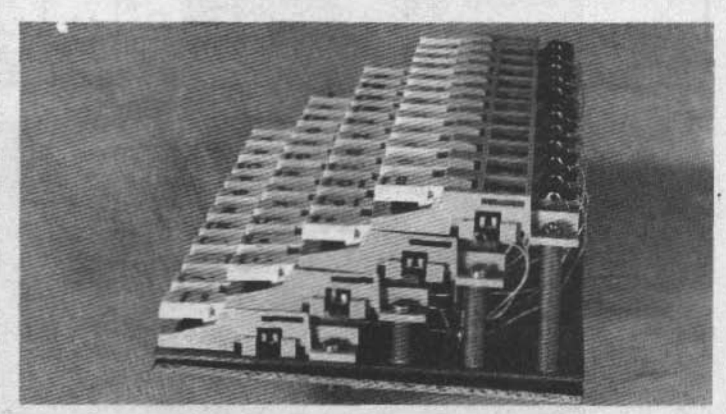
Wie schließen wir die Tastatur an?

Als Tasten eignen sich metallische und Elastomer-Kontakte. Hall-Taster reagieren nicht schnell genug. Das Betriebssystem unseres Computers setzt eine matrixförmige Anordnung voraus. Der 1-aus-16-Dekoder MH 74154 erzeugt die nötigen Auswahlsignale zum Aktivieren einer Spalte aus den untersten Adreßbits AØ bis A3 **(Abb. 35)**. Die Ausgangssignale SØ bis S3 der Tastatur werden den Eingängen der Prozessorplatine zugeführt. Jede Taste verbindet, sofern betätigt, ein Auswahlsignal (Spalte) mit einem Ausgang (SØ bis S3, Zeile).

Das Zuordnen der zugehörigen Zeichencodes (ASCII)[26] erfolgt programmtechnisch mittels der Tabelle auf den Adressen %ØFØØ bis %ØF3F im EPROM 1. Durch Ändern dieses Speicherbereichs kann jede beliebige Tastenanordnung innerhalb der maximal 15×4-Matrix realisiert werden (Pin 1 des MH 74154 muß frei bleiben).

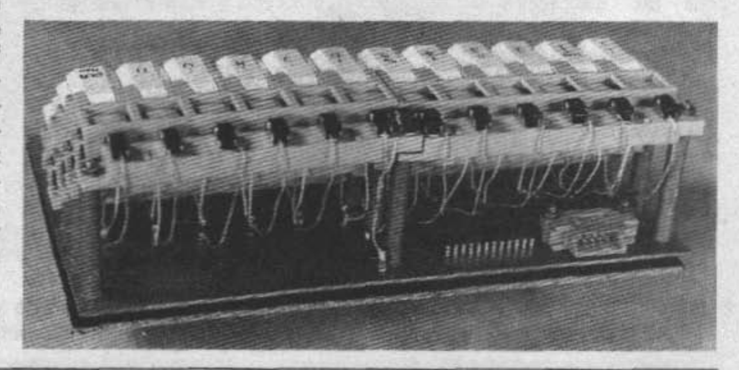
---

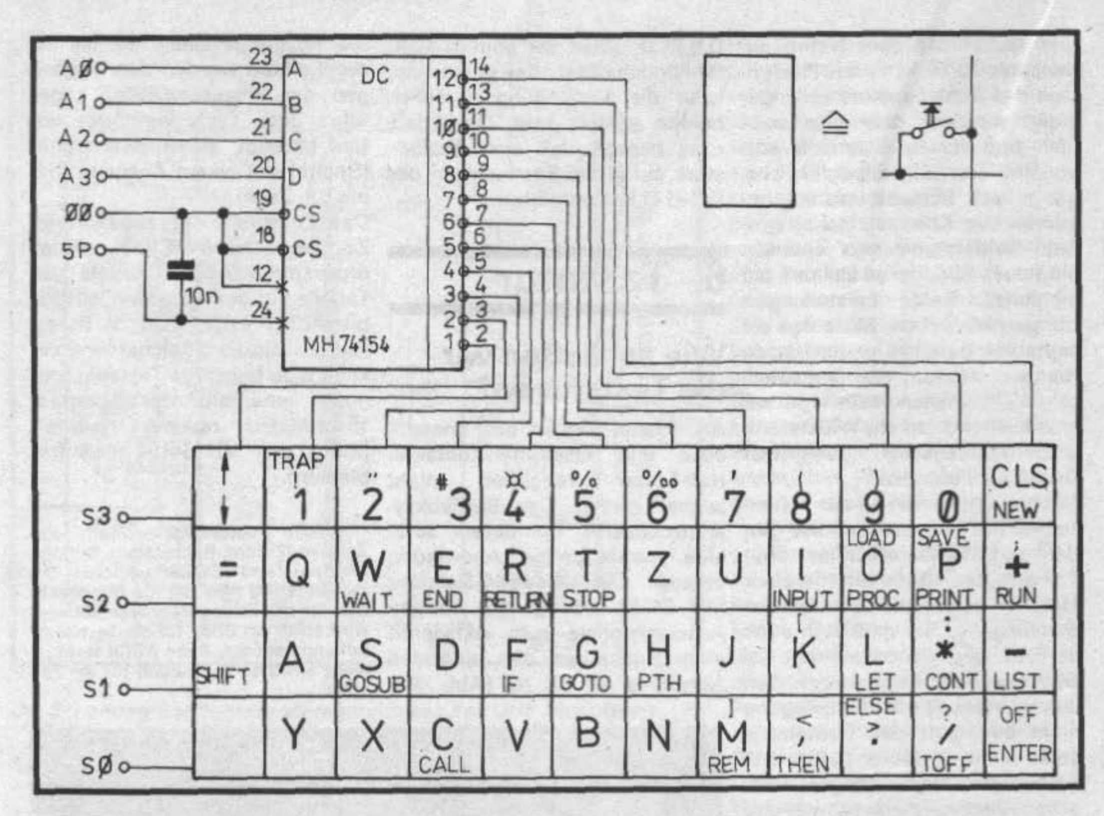
26) **ASCII** Zuordnungsvorschrift von Zeichen (Ziffern, Buchstaben, Sonderzeichen) und Steuerfunktionen zu Zahlen (siehe Abb. 68). Da Mikrorechner nur mit Zahlen operieren können, wird solch ein Code für die Textverarbeitung benötigt. Beim ASCII steht jeweils ein Byte (8-bit-Zahl) für ein Zeichen.



Stufenförmige Anordnung der TT-Tastenpulte

Verdrahtung der Tastatur



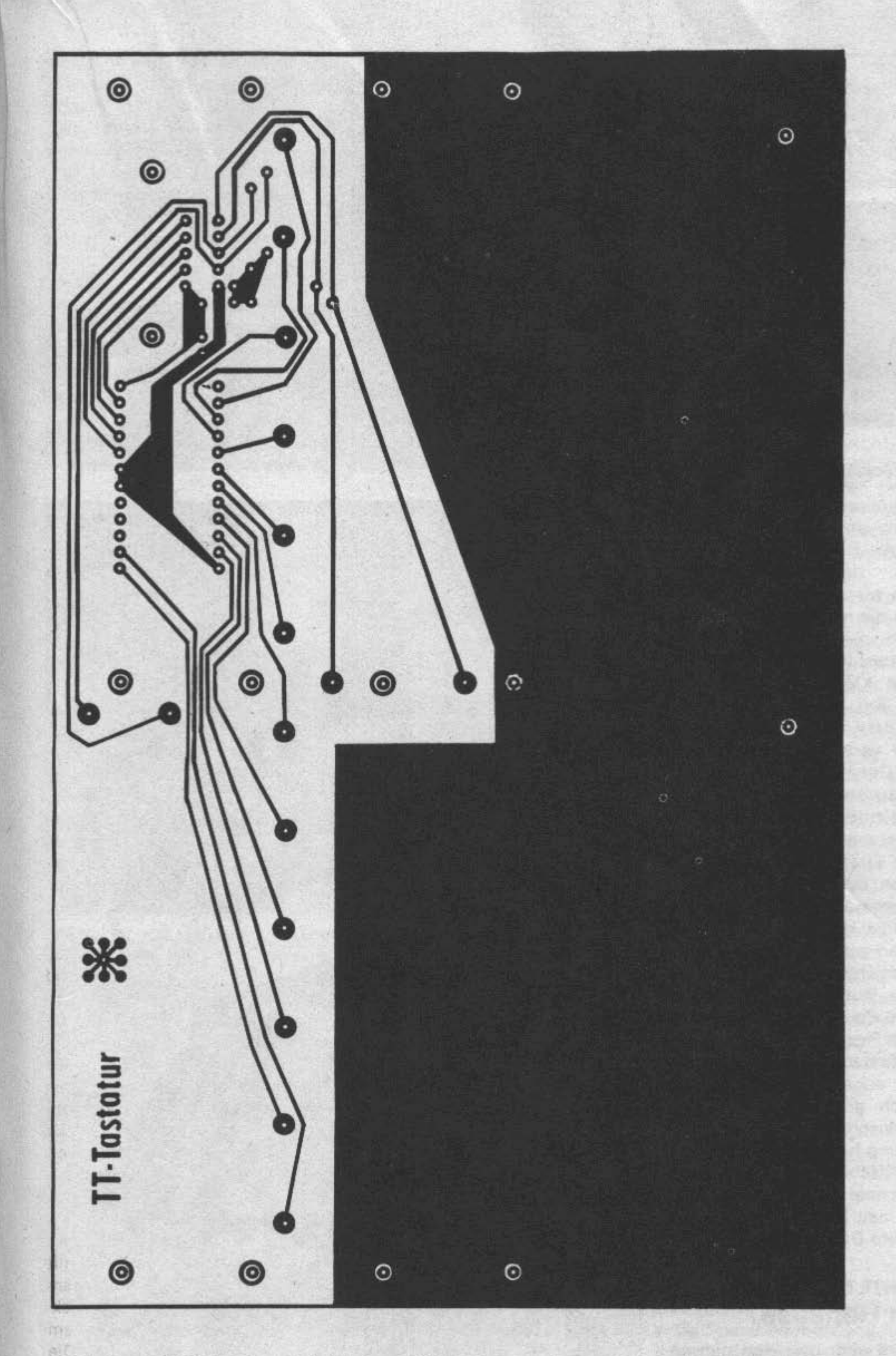


35 Schaltbild der Tastatur

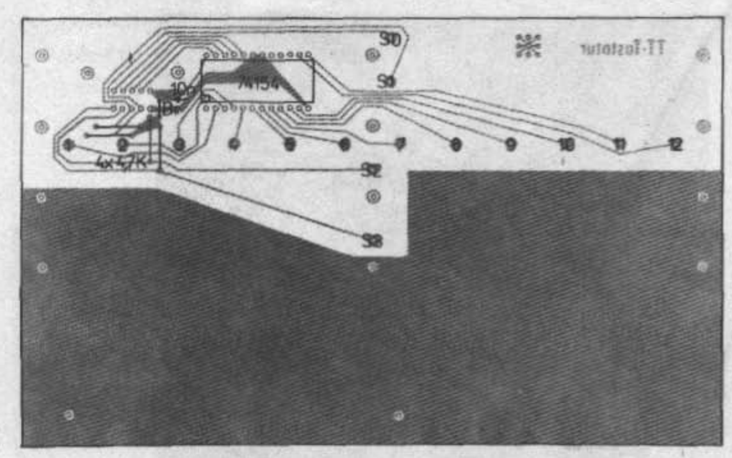
Die Anschlüsse der Prozessorplatine befinden sich auf der linken Seite (vgl. Abb. 7). Dem U 40098 D liegt S3 am nächsten, es folgen S2, S1 und SØ. Unten geht es mit AØ bis A3, Masse und 5P weiter. Da der Dekoder der Tastatur zugeordnet ist, kann sie auch abgesetzt realisiert werden. Der Verbindung dient ein 10poliges Kabel. Längen über 50 cm erfordern zusätzliche pull-up-Widerstände zwischen 5P und SØ bis S3 an der Tastatur (4×1 kΩ bis 4,7 kΩ). Um Störstrahlungen der ständig bewegten Signale AØ bis A3 zu vermeiden, muß bei abgesetzter Tastatur abgeschirmtes Kabel verwendet werden.

Wie wird die Tastatur aufgebaut?

Die Konstruktion der Tastatur bestimmt die des Computer-Gehäuses entscheidend mit. Am wenigsten Platz beansprucht eine Flachfolientastatur. Sie gestattet das Unterbringen aller Bauteile auf geringstem Raum. Den größten Platz benötigt ein aus vielen Einzeltasten (TSS oder TSE) zusammgengesetztes Tastenpult. Es bietet den größten Komfort, verursacht aber auch große Kosten (200 bis 300 Mark). Entschieden billiger und nachbausicher ist das Zusammensetzen von acht TT-Tastenpulten, die für 4 Mark je Stück im Modelleisenbahnhandel angeboten werden. Zwei davon bilden zusammengesteckt eine Zeile unserer Tastatur. Die Zungen aus Federstahl führen die Auswahlsignale, das Blech mit den Gegenkontakten realisiert einen Ausgang SØ, S1, S2 oder S3. Die zugeordneten Klemmanschlüsse befinden sich hinten oben an den Tastenpulten. Die vier Paare von Tastenpulten werden stufenförmig angeordnet. Das geschieht durch Anschrauben auf einer Leiterplatte **(Abb. 36 und 37)**. Dazu brauchen wir Distanzstücke. Die stufenförmige Anordnung läßt die Befestigungslöcher unter der linken Taste nur in der untersten Stufe nutzen. Um eine ausreichende Stabilität zu erzielen, müssen wir in den oberen Etagen einen anderen Befestigungspunkt schaffen. Dazu läßt sich der Gegenkontaktanschluß des jeweils linken Pultes eines Paares entfernen und durch ein Bohrloch Ø 3,2 mm ersetzen. Da sich die Bleche zusammengesteckter Pulte berühren, reicht der Anschluß des rechten für das Verdrahten aus. Bei dieser Konstruktion benötigen wir folgende Distanzstücke mit einem Innendurchmesser von 3 mm bis 4 mm:
2 × 12,5 mm, 1 × 17,5 mm,
2 × 25 mm, 1 × 30 mm,
2 × 37,5 mm, 1 × 42,5 mm.
Die Montage auf der bereits mit dem Dekoder und dem Kondensator (und ggf. Lötösen) bestückten Leiterplatte beginnt mit dem linken Pult der unteren Stufe. Die M-3-Schrauben oder -Gewinde-



36 Leiterzugbild der Tastaturleiterplatte für TT-Tastenpulte



37 Bestückungsplan der Tastaturleiterplatte

dem Gehäuse zu beginnen. Dann hat alles seinen festen Platz, wodurch Unfälle und technische Schäden vermieden werden. Wegen der Schirmwirkung bieten sich Metallkonstruktionen an. Plastgehäuse haben zusätzlich den Nachteil, sehr wenig Wärmeaustausch zwischen Geräteinnerem und der Umgebung zuzulassen.

Zwei bereits entsprechend zugeschnittene und gebogene Bleche reichen aus. Wichtig ist, alle Leiterplatten sicher isoliert zu befestigen und den Berührungsstifte müssen ggf. gekürzt werden, um den weiteren Aufbau nicht zu behindern. Ist eine Ebene zusammengeschraubt, müssen wir sie bereits anschließen, da die nächste die Kontakte verdeckt. Die Drahtstärke muß mindestens 0,5 mm betragen, damit die Klemmkontakte sicher halten. Bei eventuell erforderlichem Löten an den Tastenpulten ist sehr vorsichtig zu verfahren, da die Gehäuse aus Thermoplaste bestehen.

Die Beschriftung kann mit Abreibebuchstaben vorgenommen werden. Einfacher ist das Aufkleben der auf der vierten Umschlagseite gedruckten Felder. Die nötige Widerstandsfähigkeit erreichen wir durch anschließendes vorsichtiges Auftragen von Klarlack (zum Beispiel Abdecklack aus dem Ätzset) mit einem weichen Pinsel.

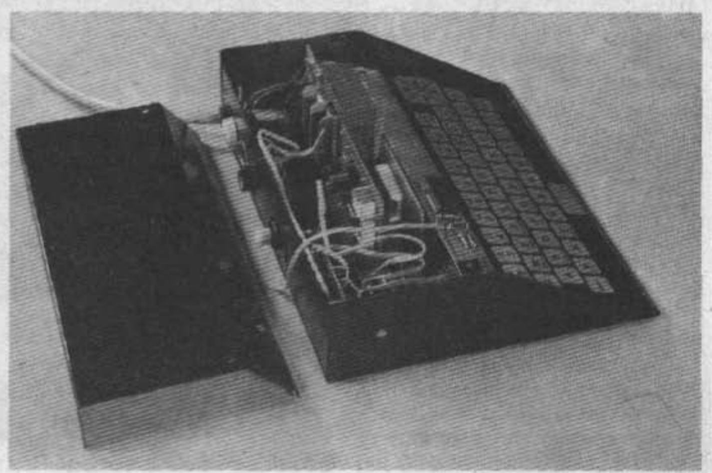
Ist die Tastatur an der Prozessorplatine angeschlossen, können wir nach gründlicher Sichtkontrolle einschalten. Jede Tastenbetätigung hat eine Reaktion auf dem Bildschirm zur Folge. Baut der Rechner das Anfangsmenü ständig neu auf, hat mindestens eine Taste Dauerkontakt.

Kommt nun alles in ein Gehäuse?

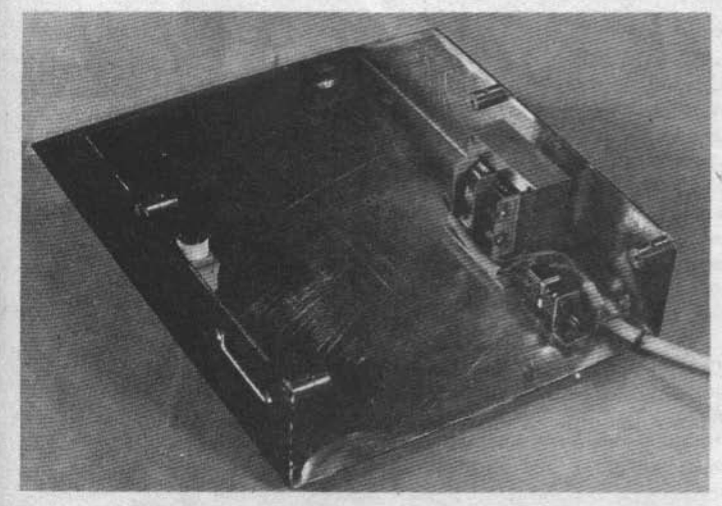
Es macht sich gut, den ganzen Selbstbau mit der Tastatur und

Grundgerät mit Folien-Flachtastatur und Standard-Peripherie

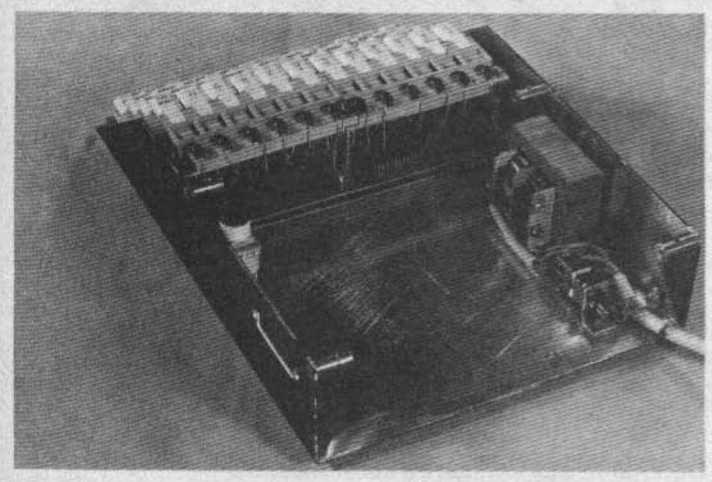




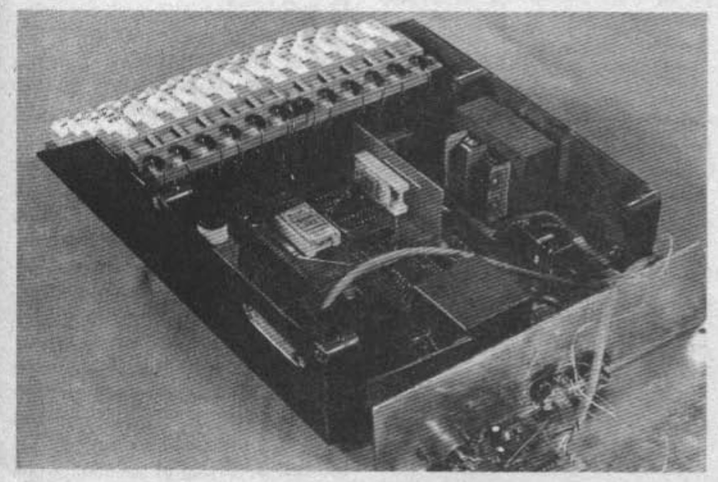
Eine flache Tastatur gestattet den Aufbau auf kleinstem Raum.



Das Gehäuse läßt sich unkompliziert konstruieren. Metallkonstruktionen bieten sich an. Zwei entsprechend zugeschnittene und gebogene Bleche reichen aus.



Günstig ist, mit Gehäuse und Tastatur zu beginnen, so hat alles gleich seinen Platz.



Das komplette Grundgerät: Der erste Speichermodul wurde wegen der Bauhöhe waagerecht angeordnet.

schutz zu gewährleisten. Wie der Trafokern müssen alle metallischen Gehäuseteile und die Signalmasse (00) eine gute Verbindung zum Schutzleiter haben. Diese Maßnahme lassen wir uns sicherheitshalber von einem Fachmann abnehmen oder realisieren. Die Wärmeableitung vom Spannungsregler B 3170 V kann, da sie nur 5 W ausmacht, auch mit Hilfe von Blechteilen des Gehäuses realisiert werden. Die Isolierschicht dürfen wir natürlich nicht vergessen, denn sonst würde die Betriebsspannung 5P kurzgeschlossen.

Die Bauhöhe der TT-Tastatur paßt gut zu der eines Heiztrafos (M55-Kern, zum Beispiel 8,3 V/1,3 A). Bei der entsprechenden Gehäusehöhe von 60 mm (Innenmaß) kann die Modulvariante A nicht senkrecht zur Prozessorplatine montiert werden. Bei paralleler Anordnung verdeckt sie die Plätze 2 und 3. Für den Speichermodul mit dem zweiten EPROM läßt sich noch der Platz 4 nutzen. In diesem Fall darf für das Signal $\overline{\text{CSB}}$ (Freigabe des EPROM) kein Drähtchen als Teil des Kontaktkamms eingelötet werden. Statt dessen erhält der Modul dieses Signal über einen Schaltdraht vom $\overline{\text{CSB}}$-Lötauge des Platzes 2. Damit belegt dann EPROM 2 den richtigen Adreßbereich (%2ØØØ bis %27FF), obwohl er auf Modulplatz 4 sitzt.

7. Bedienungsanleitung

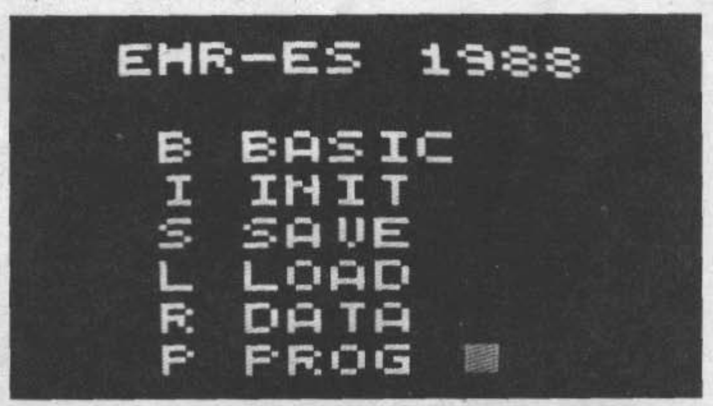
Wie bedienen wir unseren Computer?

Die Arbeit mit dem Computer ist als Dialog organisiert. Für das „Gespräch" dient die Tastatur der Informationsübertragung zum Rechner, während der Bildschirm die entgegengesetzte Richtung realisiert. Erwartet der Computer eine Tastenbetätigung, erscheint der Kursor (flimmerndes Rechteck) auf dem Bild. Nach Tastenbetätigung wird er mit dem der Taste zugeordneten Zeichen ersetzt. Das ist die Bestätigung des Rechners, die aber bei der Eingabe von Kommandos nicht immer auf dem Bild stehen bleibt.

Einigen Tasten sind gleich zwei Zeichen zugeordnet. Die Grunddeutung (groß, unten, vgl. **Abb. 35**) gilt, solange nichts anderes vereinbart ist. Nach Betätigen der Umschalttaste **(SHIFT)** erscheint rechts unten auf dem Bild ein weißer Balken. Er kennzeichnet das Aktivieren der Zweitbelegung. Der als nächstes betätigten Taste wird damit das Zeichen zugeordnet, was etwas kleiner über der Grunddeutung steht (Abb. 35).

Zwei weitere Tasten neben SHIFT haben eine besondere Funktion: **CLR** (clear) löscht das zuletzt eingegebene Zeichen und rückt den Kursor nach links. **ENTER** bestätigt eine Eingabe. Diese Taste markiert bei BASIC das Ende einer Zahlenangabe oder einer Programmzeile. Die Zweitbelegung von ENTER ist **OFF**. Sie erklärt eine Eingabe als ungültig. Bei den Betriebssystem-Komponenten DATA und PROG dient sie auch als Abbruchkommando.

Alle anderen klein geschriebenen Wörter auf den Tasten bezeichnen BASIC-Kommandos und -Anweisungen. Diese Bedeutung gilt entsprechend, wenn im Kursor ein K (Kommando) oder ein A (Anweisung) zu erkennen ist. Die



Das Anfangsmenü: sechs verschiedene Dienste



Der BASIC-Editor erwartet ein Kommando.

oben stehenden Bezeichnungen (z. B. ELSE) sind als Zweitbelegung nach SHIFT gültig.

Die Rücksetztaste RESET gehört nicht zur Tastenmatrix. Sie hat, wie wir bereits kennengelernt haben, eine besondere Funktion. Nach der Betätigung dieser Taste beginnt das Betriebssystem mit der Anzeige des Anfangsmenüs, gleichgültig, in welchem Zustand der Computer vorher war. (Der RAM-Inhalt bleibt allerdings unverändert.) Nun können wir mit den Tasten B, I, S, L, R und P zwischen sechs Angeboten wählen.

Was bedeutet BASIC?

BASIC bezeichnet eine einfache Programmiersprache, die besonders für Anfänger geeignet ist (**B**eginners **A**ll purpose **S**ymbolic **I**nstruction **C**ode). Sie enthält dem Computer verständliche Kommandos und Anweisungen in Form von aus dem Englischen abgeleiteten kurzen Wörtern. Unser Computer benutzt eine einfache Version (TINY-MP-BASIC), für die der Einchip-Mikrorechner UB 8830 D im inneren ROM einen Interpreter (Übersetzer) enthält. Im Vergleich zum in Klein- und Personalcomputern gängigen extended-BASIC gibt es hier weni-

ger Anweisungen und Kommandos. Gerechnet wird mit ganzen Zahlen im Bereich von − 32 767 bis + 32 767 (16-Bit-Festkommaformat)[27]. Gebrochene Zahlen kann der TINY-MP-BASIC-Interpreter nicht verarbeiten.

Betätigen wir bei Anzeige des Anfangsmenüs die Taste B, aktiviert unser Computer den BASIC-Editor. Seine Hauptfunktion besteht in der Eingabe von BASIC-Programmen über die Tastatur. Außerdem bietet er einige Kommandos. Er meldet sich mit der Überschrift COMP JU+TE R (einer Symbiose des Kürzels JU+TE der Zeitschrift JUGEND+TECHNIK und des Wortes Computer) und erwartet ein Kommando, was ein K im Kursorfeld kennzeichnet. Zu Beginn ist das Kommando **NEW** (neu, Taste CLR) üblich, das den BASIC-RAM für eine Neueingabe löscht. Wir können das ausprobieren. Der Kursor rutscht zurück, läßt das K stehen. Das entspricht der Tastenfunktion CLR. Nach ENTER entsteht wieder das ursprüngliche Bild.

Nun können wir den BASIC-Interpreter nutzen, um alle Tasten auszuprobieren. Dazu betätigen wir erst die Taste 1, dann ENTER, dann P, dann SHIFT und 2. Auf dem Bildschirm steht jetzt:

1

PRINT"

Im folgenden erscheint jede Tastenbetätigung auf dem Bildschirm. Nur " (SHIFT 2) und ENTER dürfen zunächst nicht noch einmal eingegeben werden. Wenn wir diese Folge zum Schluß anhängen, haben wir bereits unser erstes Programm geschrieben.

Wieso?

Die Eingabe einer Programmzeile beginnt mit der Zeilennummer (1), als deren Abschluß der Interpreter die ENTER-Taste erwartet. Es folgt die Eingabe einer Anweisung. Wer aufgepaßt hat, weiß, daß an dieser Stelle ein A im Kursorfeld stand. Der Taste P ist die PRINT-Anweisung zugeordnet. Das heißt eigentlich drukken, dient aber der Anzeige auf dem Bildschirm. Es kann zum Beispiel ein beliebiger Text angezeigt werden, der in Anführungszeichen einzuschließen ist. Genau das haben wir getan und zuletzt auch noch die Eingabe der Programmzeile mit ENTER abgeschlossen.

Damit steht nun die Programmzeile 1 mit der PRINT-Anweisung im RAM. Das Kommando **RUN** (Lauf; Taste +) läßt sie vom Interpreter ausführen. Dabei erscheint alles noch einmal auf dem Bildschirm wie beim Ausprobieren der Tasten. Zum Schluß bringt der Computer die Fehlerausschrift ERROR Ø, weil unser Programm kein ordentliches Ende hat. Er nimmt alles sehr genau. Geben wir erneut das Kommando RUN, geschieht das gleiche noch einmal.

Welche Kommandos versteht der Editor?

Kommandos führt der Editor sofort aus. **Abb. 38** gibt einen Überblick. NEW und RUN kennen wir bereits. Wichtig ist auch **LIST.** Solange nach K-Kursor die Taste − bestätigt wird, zeigt der Editor das im RAM stehende Programm Zeile für Zeile an. **CONT** (Taste *) brauchen wir beim Testen umfangreicher Programme. Dieses Kommando setzt mit der STOP-Anweisung unterbrochene Programme fort. Es wirkt wie RUN, nur daß nicht ab der ersten Zeile, sondern mit der nach STOP stehenden Anweisung gestartet wird.

Abb. 38: BASIC-Kommandos (bei K-Kursor)

NEW	Löschen des BASIC-RAM für die Neueingabe von Programmen
RUN	Start des BASIC-Programms
LIST	Anzeige des BASIC-Programms
CONT	Programmfortsetzung nach STOP
SAVE	Speichern des BASIC-Programms auf Kassette
LOAD	Laden eines BASIC-Programms von Kassette
Ziffer	Eingabe einer Programmzeile
Buchstabe	Anzeige des betreffenden Variableninhalts

Für die Arbeit mit dem Kassettenrecorder benötigen wir die Kommandos SAVE (sichern) und LOAD (laden). Beide sind als Zweitbelegung über SHIFT zu erreichen. Nach **SAVE** (SHIFT P) löscht der Editor den Bildschirm und erwartet die Eingabe des Programmnamens. Nach Betätigen der ENTER-Taste beginnt die Ausgabe über die Interface-Leiterplatte. Dem Vorton (der bereits mit aufgenommen werden muß) folgen der erste Block mit Anfangs- und Endadresse des BASIC-Programms, die der Editor selbst ermittelt, und den ersten elf Buchstaben des eingetasteten Programmnamens. Dann gibt der Computer je nach Programmlänge einen oder mehrere Datenblöcke mit je 128 Byte aus. Wenn zum Schluß wieder eine Anzeige auf dem Bildschirm erscheint, kann der Kassettenrecorder gestoppt werden.

Das Aufzeichnen von Daten erfordert einen gut funktionierenden Recorder und einwandfreies Bandmaterial. Am besten eignen sich die Geräte aus dem VEB Elektronik Gera (Geracord, LCR) und CrO_2-Kassetten. Bei Handaussteuerung ist es günstig, etwas zu laut aufzuzeichnen, da so Ungleichmäßigkeiten der Magnetschicht (drop outs) besser ausgeglichen werden. Wir sind gut beraten, wichtige Programme gleich zweimal hintereinander aufzunehmen. Das gibt eine größere Sicherheit.

Zum Laden von Programmen in den Computer muß die Kassette zuerst auf den Anfang der Aufzeichnung gespult werden. Nach Einschalten der Wiedergabe starten wir mit **LOAD** (SHIFT O) das Ladeprogramm, das wie SAVE ohne Bildschirm arbeitet. Er bleibt also dunkel. Ob es vor oder

[27] **Festkommaformat** Rechnerinterne Zahlendarstellung mit fester Zuordnung der Stellenwerte zu den Bitpositionen auf der Basis des dualen Zahlensystems

während des Vortones beginnt, ist gleichgültig. Gelingt das Laden, erscheinen FF (file found) und der gelesene Programmname auf dem Bildschirm. Im Fehlerfall wird nur die Nummer des nicht lesbaren Blocks angezeigt. Wie bei den Kleincomputern KC 85 und KC 87 prüft das Ladeprogramm die Blocknummernfolge und die zu jedem Block gehörenden Kontrollsummen.

Wird bei K-Kursor eine **Zifferntaste** betätigt, beginnt die Eingabe einer Programmzeile. Im Gegensatz zu anderen Editoren muß als Abschluß der Zeilennummer und als Zeilenende die ENTER-Taste betätigt werden. Damit entsteht die Eingabevorschrift:

Zeilennumer ENTER Anweisung ... ENTER

Zum Testen von Programmteilen können STOP-Anweisungen eingefügt werden. Erreicht der Interpreter diese Anweisung, kehrt er in den Kommandomodus zurück. Bevor mit CONT fortgesetzt wird, können wir uns durch das Betätigen der entsprechenden **Buchstabentaste** zur Kontrolle die Variableninahlte anzeigen lassen. Das gestattet, die Wirkung einzelner Anweisungen oder ganzer Programmteile genau zu verfolgen. Für das Auffinden von Programmfehlern (Entlausen, debugging) ist dieses Verfahren eine große Hilfe. Der Editor erleichtert das Orientieren, indem er bei jeder Unterbrechung anzeigt, in welcher Zeile die betreffende STOP-Anweisung steht.

Was sind Variable?

Beim Ausführen von Programmen muß sich der BASIC-Interpreter Zahlenwerte merken können. Dazu benutzt er 26 **Variable** mit den Bezeichnungen A bis Z. Durch Anweisungen im Programm können ihnen Werte zugewiesen werden, die sie bis auf Widerruf (Neuzuweisung) speichern. Diese Werte können wir in Ausdrücken verwenden, indem wir einfach den Variablennamen (Buchstabe) notieren.

Abb. 39: Operationszeichen

+	plus
–	minus
*	mal
/	geteilt durch
□M	Divisionsrest
□A	UND (AND)
□O	ODER (OR)
□X	ANTIVALENZ (XOR)

Ausdrücke sind ganze Zahlen, Variable, Funktionen oder Verknüpfungen von Zahlen, Variablen und Funktionen. Die Verknüpfungen werden durch Operationszeichen formuliert **(Abb. 39)**. Sie realisieren arithmetische Berechnungen, also die vier Grundrechenarten, den Divisionsrest und die logischen Verknüpfungen UND, ODER und ANTIVALENZ. Der Interpreter behandelt alle diese Operationen gleichwertig. Verkettete Verknüpfungen berechnet er streng von links nach rechts. Mit dem Setzen von runden Klammen können wir aber Prioritäten einführen, das Eingeklammerte wird zuerst berechnet und dann in die Bearbeitungsfolge eingereiht. Bei der Programmausführung ermittelt der Interpreter mit den betreffenden Verknüpfungen den Wert eines Ausdrucks, der dann der Anweisung entsprechend verwendet wird.

Funktionen (Abb. 40) gestatten, Eingaben (GTC, INPUT), einfache Manipulationen eines Operanden (ABS, NOT, RR, RL) sowie Inhalte von Registern (GETR, GETRR) und Speicherzellen (GETEB, GETEW) in Ausdrücke aufzunehmen. Weil im TINY-MP-BASIC keine gebrochenen Zahlen verarbeitet werden, fehlen die Winkelfunktionen.

Die Eingabe von **Zahlen** erfolgt dezimal mit ggf. vorangestellten Minuszeichen. Nach % dürfen Hexadezimalzahlen (Ziffern: Ø bis 9 und A bis F) formuliert werden. Die genannten Regeln erlauben beliebig lange Ausdrücke mit vielen aneinandergereihten Operationen. Es dürfen auch praktisch unbegrenzt Verschachtelungen mit runden Klammern formuliert werden. Hauptsache: jede Eröffnung hat auch ihren Abschluß. Nur bei Verschachtelungen von Funktionen untereinander streikt der Editor. Wie die Programmbeispiele im dritten Teil der Broschüre zeigen, werden in praktischen Programmen fast immer einzelne Zahlen oder Variable, seltener einfache Verknüpfungen und fast nie längere Ketten als Ausdruck verwendet.

Ausdrücke sind immer Zahlenwerte. Bei der Ausgabe auf den Bildschirm können auch Texte verarbeitet werden. Man bezeichnet sie mit **Zeichenkette.** Sie beginnen mit einem Anführungszeichen, dem der darzustellende Text folgt. Den Abschluß bildet wieder ein Anführungszeichen. Wir haben das bereits probiert. Der Text darf alle Zeichen (Buchstaben, Ziffern und Sonderzeichen) enthalten. Ausgenommen sind nur das Anführungszeichen und ENTER.

Welche Anweisungen gibt es?

Anweisungen werden mit Schlüsselworten kodiert, die der

Abb. 40: Funktionen

ABS [Ausdruck]	Betrag des Ausdrucks
NOT [Ausdruck]	logisches Komplement des Ausdrucks
RR [Ausdruck]	Ausdruck rechtsverschoben
RL [Ausdruck]	Ausdruck linksverschoben
GTC	Tastencode (ASCII) im niederen Byte
INPUT	Eingabewert
GETR [Ausdruck]	Inhalt des Registers mit der Adresse Ausdruck
GETRR [Ausdruck]	Inhalt eines Doppelregisters
GETEB [Ausdruck]	Inhalt einer Speicherzelle
GETEW [Ausdruck]	Inhalt einer Doppelspeicherzelle

Bei der Eingabe von Funktionen erfolgt keine Syntaxprüfung.

Abb. 41: BASIC-Anweisungen

WAIT Ausdruck	Warten mit Ausdruck × 7 ms Dauer, Ausdruck > 0
END	Programmende
RETURN	Rücksprung ins Hauptprogramm (Umkehrung von GOSUB)
STOP	Programmunterbrechung
INPUT Zeichenkette Variable	Zuweisung eines Variablenwertes per Tastatur, Zeichenkette kann entfallen
PROC Prozedur	Ausführung einer Prozedur **(vgl. Abb. 42)**
PRINT Zeichenkette Ausdruck	Anzeige von Zeichenkette und Ausdruck dezimal, Zeichenkette und Ausdruck können entfallen
GOSUB Ausdruck	Unterprogrammsprung zu Zeile Nr. Ausdruck
IF Vergleich THEN Anweisung	Bedingte Ausführung von Anweisung, nur wenn Verleich **(vgl. Abb. 43)** erfüllt
GOTO Ausdruck	Sprung zu Zeile Nr. Ausdruck
PTH Zeichenkette Ausdruck	Anzeige von Zeichenkette und Ausdruck hexadezimal, Zeichenkette und Ausdruck können entfallen
LET Variable = Ausdruck	Zuweisung eines Variablenwertes, weitere Zuweisungen können mit Komma getrennt folgen
CALL Ausdruck	Aufruf eines Maschinenunterprogramms mit Startadresse Ausdruck
REM beliebiger Text	Kommentar ohne Wirkung auf den Programmablauf
ELSE; Anweisung	Bedingte Ausführung von Anweisung, nur wenn Bedingung in voriger Zeile nicht erfüllt
TRAP Vergleich, Ausdruck	Falle, vor dem Ausführen jeder folgenden Zeile wird Vergleich getestet und, falls erfüllt, mit GOSUB zu Zeile Nr. Ausdruck verzweigt und Falle gelöscht
TOFF	Löschen der Falle

Abb. 42: Prozeduren:

PTC [Ausdruck]	Anzeige von Ausdruck gemäß ASCII
SETR [Ausdruck 1, Ausdruck 2]	Laden des Registers Nr. Ausdruck 1 mit dem Wert von Ausdruck 2
SETRR [Ausdruck 1, Ausdruck 2]	Laden eines Doppelregisters
SETEB [Ausdruck 1, Ausdruck 2]	Laden einer Speicherzelle
SETEW [Ausdruck 1, Ausdruck 2]	Laden einer Doppelspeicherzelle

Bei der Eingabe von Prozeduren erfolgt keine Syntaxprüfung.

Abb. 43: Vergleich

Ausdruck > Ausdruck	größer als
Ausdruck < Ausdruck	kleiner als
Ausdruck = Ausdruck	gleich
Ausdruck >= Ausdruck	größer oder gleich
Ausdruck <= Ausdruck	kleiner oder gleich
Ausdruck <> Ausdruck	ungleich

Editor mit dem A-Kursor erwartet. Deren Eingabe bedarf nur einer Tastenbetätigung, im Bild erscheint das vollständige Schlüsselwort. Nur THEN kann ohne A-Kursor eingegeben werden. Anweisungen speichert der Computer zunächst nur. Erst nach dem RUN-Kommando im Rahmen der Programmabarbeitung führt er sie aus. Statt einer Anweisung gestattet der Editor mehrere mit Semikolon getrennte Anweisungen einzugeben. **Abb. 41** gibt eine Übersicht aller Anweisungen, wobei wie in Abb. 40 auch die Formulierungsvorschrift (Syntax) eingearbeitet ist.

Um ein bißchen Klarheit in die vielen Regeln zu bringen, wollen wir nun Schritt für Schritt ein einfaches Programm mit den wichtigsten Anweisungen entwerfen. Am häufigsten ist die Zuweisung **LET** (Taste L). Der Interpreter berechnet bei Ausführung dieser Anweisung zuerst den Ausdruck und speichert dann das Ergebnis in der betreffenden Variablen. Er deutet diese Anweisung also nicht als Gleichung! Nach dem Kommando NEW können wir zum Beispiel die folgende Programmzeile eingeben:

```
1Ø LET A=4
```

Das Numerieren in Zehnerschritten ist üblich, um Raum für nachträgliche Ergänzungen zu lassen. Der Ausdruck besteht hier nur aus der Zahl 4. Nach dem zweiten ENTER erwartet der Editor wieder ein Kommando. Mit RUN läßt sich unser Minprogramm starten. Der Editor moniert wieder mit ERROR Ø die fehlende END-Anweisung, aber mit Betätigen der Taste A (bei K-Kursor) können wir uns von der Ausführung überzeugen.

Erscheint die Zuweisungsvariable (hier: A) auch im Ausdruck der LET-Anweisung, geht der alte Inhalt in die Rechnung ein, während dann das Ergebnis als neuer Inhalt gespeichert wird. Mit

```
2Ø LET A=A+1
```

erhält der Interpreter die Anweisung, A um 1 zu erhöhen. Nach RUN und A zeigt uns der Editor das neue Ergebnis (5). Mit LIST können wir unser nun zweizeiliges Programm ansehen. Zum Vermeiden der Fehlermeldung eignet sich die **END**-Anweisung (Taste E). Sie bedarf keiner zusätzlichen Angaben:

```
1ØØ END
```

Nun meldet der Editor nach RUN ein ordnungsgemäßes Programmende mit Zeilennummer. Zwischen 2Ø und 1ØØ bleibt viel Platz für weitere Anweisungen. Sinnig ist hier eine automatische Ergebnisanzeige mit **PRINT**, die uns vom nachträglichen Holen des Rechenergebnisses befreit. Auf dem Bildschirm erscheinen der Text der Zeichenkette und der Wert des Ausdrucks. Zeichenkette und Ausdruck können auch entfallen. Wir ergänzen unser Programm mit:

```
3Ø PRINT A
```

Nach dem Ausdruck dürfen, mit Komma getrennt, weitere Zeichenketten und Ausdrucke folgen, um Maßeinheiten und weitere Werte in der gleichen Bildschirmzeile anzuzeigen. Da eine

Dezimalzahl mit Vorzeichen sechs Zeichenpositionen braucht, bleibt dafür nicht viel Platz bei 13 Zeichen je Zeile. Nach Ausführung einer PRINT-Anweisung setzt der Interpreter den Kursor auf den Anfang der nächsten Bildschirmzeile, wenn er sich nicht schon links befindet. Mit LIST können wir uns anschauen, wie der Editor die Anweisung einsortiert hat. Nach RUN erhalten wir unser Resultat nun auch angezeigt. Mit

3Ø PRINT "A="A

läßt sich eine kurze Erklärung der Anzeige ergänzen. Bei der Eingabe einer Programmzeile streicht der Editor eine eventuell schon vorhandene mit dieser Nummer. Das + in der Anzeige stammt vom RUN-Kommando, mit

5 PRINT

beginnt unsere erste Ausgabe in einer neuen Bildschirmzeile.

Unser Programm besteht inzwischen aus fünf Zeilen. Sie kommen in der Reihenfolge, in der sie der Editor einsortiert hat, nach RUN zur Ausführung. Abweichende Abarbeitungsfolgen kann man mit den Programmsprung **GOTO** (gehe zu, Taste G) erreichen. Der Interpreter setzt nach dieser Anweisung mit der als Ausdruck angegebenen Zeilennummer fort. Gewöhnlich steht eine Zahl als Ausdruck.

4Ø GOTO 2Ø

Mit dieser Anweisung mündet unser Programm in eine endlose Schleife. Nach RUN erhöht der Interpreter ständig die Variable A und zeigt sie an. Er läßt sich nur noch mit der RESET-Taste stoppen. Das Programm und die Variableninhalte bleiben dabei erhalten. Mit den Tasten A und LIST können wir das überprüfen.

Es gibt auch die Möglichkeit, eine oder mehrere Anweisungen nur unter bestimmten Bedingungen ausführen zu lassen. Das Kodieren wir mit den Schlüsselworten **IF** (falls, Taste F) und **THEN** (dann, Taste ,). Die Bedingung ist durch den Vergleich festgelegt. Mit den Operatoren < (kleiner), > (größer) und = (gleich) werden zwei Ausdrucke verglichen. In unser Programm paßt:

4Ø IF A<5Ø THEN GOTO 2Ø

Nach RUN erhöht der Interpreter nun A nur noch bis 5Ø und findet dann das Programmende. Wegen der nicht mehr erfüllten Bedingung A<5Ø ignoriert er dann die GOTO-Anweisung.

LET ist nicht die einzige Möglichkeit, einer Variablen einen Wert zuzuweisen. Der Interpreter gestattet auch die Eingabe per Tastatur. Dazu dient die **INPUT**-Anweisung (Taste I). Der Interpreter fordert bei deren Ausführung eine Zahleneingabe, die mit ENTER abzuschließen ist. Die Zeichenkette, die auch entfallen kann, wird gewöhnlich für die Anzeige einer Eingabeaufforderung genutzt:

1Ø INPUT "A: "A

Damit soll unser kleines Programmbeispiel, das alle wichtigen Anweisungen enthält, sein Ende finden. Es zeigt ab Eingabewert, der auch negativ sein kann, alle Zahlen bis 5Ø an. Auch die anderen Anweisungen lassen sich am besten verstehen, wenn wir sie mit so einfachen Beispielen ausprobieren.

Nun noch ein Hinweis: Zum effektiven Strukturieren von BASIC-Programmen verwendet der Editor des Computers zwei spezielle Bytes: %ØD (Zeilenende) und %ØØ (Programmende). Diese dürfen sonst nicht im BASIC-RAM erscheinen. Als niederer Teil der Zeilennummer wären sie aber denkbar. Der Editor ersetzt selbständig das Nullbyte mit %Ø1. Daher erscheint bei LIST als Zeilennummer statt Ø die 1, statt 256 die 257, statt 512 die 513 usw. Nicht verwendet werden dürfen die Zeilennummern 13, 269, 525, 781 usw., da sie %ØD als niederes Byte im dualen Zahlenformat besitzen. Die höchstmögliche Zeilennummer ist 32767.

Wann erscheinen Fehlermeldungen?

Der Editor bringt die Anzeige **ERROR** mit einer kennzeichnenden Zahl, wenn eine falsche Eingabe oder ein Ausführungsfehler erkannt wird. Das Eingeben von Programmzeilen verfolgt der Editor schritthaltend. Er macht sofort auf Syntaxfehler aufmerksam, was für Anfänger sehr hilfreich ist. Dafür dürfen hier keine Tippfehler unterlaufen, die Taste CLR ist tabu! Bei falscher Eingabe muß, auch wenn der Editor nichts moniert, noch einmal die ganze Zeile neu begonnen werden.

Beim Auftreten mehrerer Fehler erscheint die Summe der zugeordneten Zahlen. Bei Addition und Subtraktion führt die Bereichsüberschreitung nicht zum Programmstop. **Abb. 44** gibt eine Übersicht der Fehlermeldungen.

Abb. 44: Fehlermeldungen

ERROR Ø	Syntaxfehler, CONT ohne STOP, Programmende ohne STOP oder END
ERROR 1	mehr als 15 GOSUB verschachtelt
ERROR 2	RETURN ohne GOSUB
ERROR 4	Division durch Null
ERROR 8	Überschreitung des zulässigen Zahlenbereichs

Was bedeutet INIT?

Nach BASIC finden wir im Anfangsmenü INIT. Die RESET-Taste führt nur zum Neuinitialisieren des Prozesses und läßt den RAM-Inhalt unverändert. Die Eingabe eines Maschinenprogramms mit Marken[28] setzt aber gelöschten Operativspeicher voraus. Nach Betätigen von I, SHIFT und 1 aus dem Anfangsmenü wird daher der gesamte RAM mit Nullbytes geladen. Um versehentliches Löschen zu erschweren, sind drei statt sonst einer Tastenbetätigung nötig.

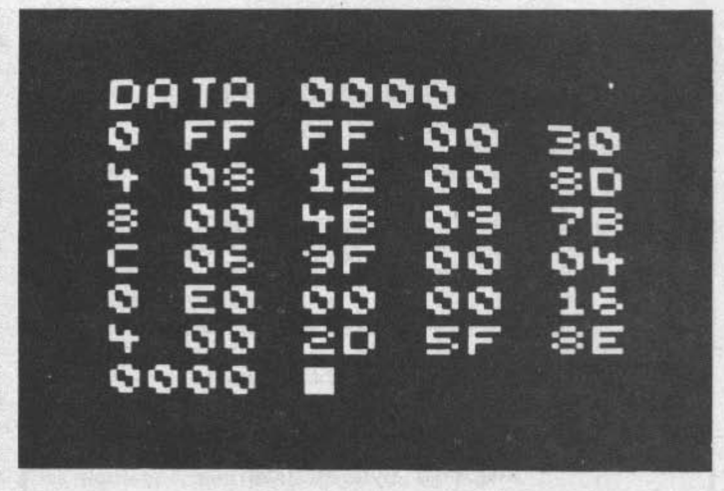
[28] **Marke** Symbolische Bezeichnung der Adresse eines Maschinenbefehls, die beim Binden für das automatische Ermitteln der Operanden von Sprungbefehlen in **Maschinensprache** benötigt wird.

Sind SAVE und LOAD BASIC-Kommandos?

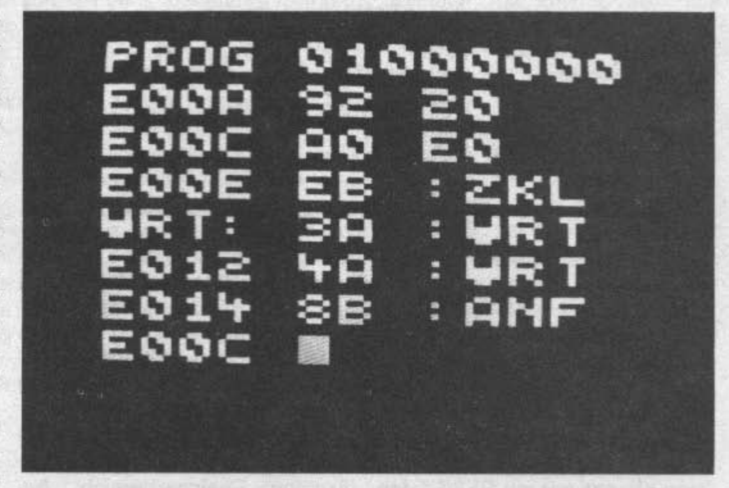
Die BASIC-Kommandos SAVE und LOAD (vgl. Abb. 38) sind mit automatischer Adreßverwaltung verbunden. Da der Nutzer bei BASIC die konkreten Speicheradressen nicht kennt, ist das dort auch sehr wichtig. Beim Programmieren in Maschinensprache verwaltet der Programmierer den Speicher und muß den Adreßbereich, der aus- oder eingelagert werden soll, selbst bestimmen. Daher enthält das Anfangsmenü im Unterschied zu den BASIC-Kommandos SAVE und LOAD die entsprechenden Angebote.

SAVE (Taste S) ist das universelle Kassetten-Ausgabeprogramm. Der Computer erwartet die Anfangs- und die Endadresse des auszulagernden Speicherbereichs bei leerem Bildschirm als vierstellige hexadezimale Eingabe **ohne** führendes % und ohne ENTER. Als drittes ist der Programm- oder Dateiname einzugeben und mit ENTER abzuschließen. Damit beginnt wie bei BASIC die Ausgabe von Vorton und Datenblöcken. Der Systembereich ab %FDØØ darf nicht mit ausgelagert werden.

LOAD (Taste L) startet noch nicht das Ladeprogramm. Zunächst bietet der Computer die Eingabe einer vierstelligen Anfangsadresse (ohne % und ENTER) an, ab der die gelesenen Daten abgelegt werden. Mit ENTER und allen anderen nicht für die Adreßeingabe tauglichen Tasten können wir auf dieses Angebot verzichten. Die Daten gelangen dann auf den Bereich, von dem sie mit SAVE geholt wurden. Das Kassettengerät darf auch hier vor oder nach Aktivieren von LOAD gestartet werden. Das Laden in den Systembereich ab %FDØØ verändert den Stapelspeicher und ist daher verboten.



Hexadezimale Anzeige der Register %ØØ bis %17 mit DATA



Teil eines Anwenderprogramms in Maschinensprache mit dem PROG-Modus angezeigt: oben rechts – die acht Flagbits, links – Adressen, z. T. als Marke vereinbart, Mitte – Operationscodes, rechts – Operanden, teilweise symbolisch notiert

Was heißt DATA?

Mit der Taste R kann aus dem Anfangsmenü oder dem PROG-Modus die RAM- und Register-Anzeige **DATA** gerufen werden. Sie zeigt ab der oben rechts stehenden Adresse 24 Bytes hexadezimal an. Adressen unter %Ø1ØØ betreffen Register, ab %Ø1ØØ Speicherzellen. In der untersten Zeile wird die Eingabe auf die angezeigte Adresse angeboten, die mit den Tasten Ø bis 9 und A bis F hexadezimal erwartet wird. Nach jeweils zwei Hexaziffern (= 1 byte) aktualisiert der Computer die Anzeige und erhöht die Eingabeadresse. Sie kann zur Korrektur mit – zurückgestellt werden. + erhöht die Anzeigeadresse um 1, ENTER um 4. Nach **G** ist die Eingabe einer neuen Anzeigeadresse vierstellig hexadezimal möglich. OFF (SHIFT ENTER) bewirkt die Rückkehr zum Anfangsmenü oder Programm-Modus. Alle anderen Tasten aktualisieren die Anzeige und stellen die Eingabeadresse

Kommandos bei DATA und PROG

0.....9 und A....F	Hexadezimaleingabe
-	Adresse -1
+	Adresse +1
Enter	Anzeige nächste Zeile
G adr	Anzeige ab adr
OFF	Rücksprung

weitere Kommandos bei PROG

: ASCII ASCII ASCII	Markeneingabe
Q	Eingabe in Zeile 1
X	Befehl streichen
I	Byte einfügen
T adr	Binden auf adr
S	Schritt-Test
N	Schleifen-Test
H adr	Haltepunkt auf adr
L	Echtzeit-Test
R	Aufruf DATA

45

auf das erste angezeigte Byte **(Abb. 45)**.

Für die Arbeit mit BASIC bietet DATA nicht nur die Möglichkeit, sich den internen Zwischencode des Programms, wie es im Speicher steht, anzusehen. Wir können auch ein versehentliches NEW-Kommando reparieren. Dazu bringen wir mit **GE000** den BASIC-RAM-Anfang zur Anzeige. Das erste Byte (oben links) ist nach NEW mit 00 belegt, sonst steht das Programm noch im Speicher. Mit der Eingabe 80 stellen wir den ursprünglichen Zustand wieder her, was sich mit OFF, B und LIST überprüfen läßt.

Wozu brauchen wir PROG?

PROG (vlg. Anfangsmenü) dient der Programmierung in Maschinensprache. Sie zu beschreiben, würde den Rahmen dieser Bauanleitung sprengen. Hierfür ist das Buch „Einchip-Mikrorechner" von Kieser/Bankel, das im VEB Verlag Technik Berlin erschien, sehr zu empfehlen. Eine populärwissenschaftliche Einführung gibt das „ABC Einchip-Mikrorechner" in den Heften 7/1988 bis 6/1989 der Zeitschrift „JUGEND+TECHNIK" aus dem Verlag Junge Welt Berlin.

Die Eingabe von Maschinenprogrammen erfolgt in einen gelöschten Speicherbereich. Besitzt der RAM keine Stützbatterie, muß deshalb nach jedem Einschalten des Computers erst INIT ausgeführt werden. Da neben dem Maschinenprogramm automatisch eine Markentabelle (Verzeichnis der symbolischen Adressen) gespeichert wird, ist mindestens die doppelte RAM-Kapazität, die das eigentliche Programm braucht, nötig. Die Markentabelle wächst in Richtung niedriger Adressen. Deshalb ist es sinnvoll, hiermit am Ende des zur Verfügung stehenden Bereichs zu beginnen.

Nach Betätigen der Taste P erwartet das Betriebssystem mit der Ausschrift **MTB** die hexadezimale Eingabe dieser Adresse. Soll der Bereich von %E000 bis %E3FF genutzt werden, erfolgt entsprechend die Eingabe E3FF. Danach erscheint die Programmanzeige ab dieser Adresse auf dem Bildschirm. Mit GE000 müssen wir sie auf den Anfang unseres Eingabebereichs stellen. Neben PROG steht in der oberen Bildschirmzeile der Inhalt des Flagregisters dual, also eine Ziffer für jeden Flagbit. Beim Programmtest spielen sie eine große Rolle, weil die ersten vier (C, Z, S, V) als Sprungbedingung dienen können. In den nächsten sechs Zeilen stehen die Adresse und der hexadezimale Code je eines Maschinenbefehls. In der untersten Zeile wird wieder die Eingabe angeboten. Sie erfolgt an die Stelle des **zweiten** Befehls. Das gestattet, sich stets an der vorherigen Eingabe, die darüber steht, zu orientieren. Zur Korrektur kann nach **Q** an die Stelle des ersten Befehls eingegeben werden.

Ähnlich wie bei DATA erzeugt der Computer nach ENTER die Anzeige ab nächstem Befehl, + und – verändern die Adresse um einen Zählschritt. Mit einem Doppelpunkt (:)beginnt die Eingabe symbolischer Adressen. So eine Marke besteht aus drei beliebigen Zeichen, die dem Doppelpunkt folgen müssen. Vor der Eingabe des Operationscodes werden linksstehende Marken eingetastet. Sie erscheinen in der Befehlsanzeige anstelle der Adreßangabe. Nach dem Operationscode lassen sich rechts stehende Marken eingeben. Das gelingt jedoch nur bei Sprungbefehlen. Sie erscheinen als Operand statt des zweiten bzw. zweiten und dritten Bytes in der Anzeige. Damit die Zuordnung der Marken zu den Befehlen nicht verlorengeht, darf man Teile eines Programms nicht durch leeren Speicherbereich (Nullbytes) trennen.

Zur Korrektur streicht **X** den aktuellen Befehl und läßt das folgende Programm aufrücken. Umgekehrt kann mit **I** (insert) ein Byte eingefügt werden. Dabei ist

zu beachten, daß das Einfügen eines Zweibytes-Befehls nach zweimal I, eines Dreibyte-Befehls nach dreimal I möglich wird. Wir sollten das Benutzen dieser Kommandos nicht scheuen, da einfaches Überschreiben das Übrigbleiben unrichtiger Markenvereinbarungen zur Folge haben kann. Beim Binden (Adreßberechnen) würde dann das Programm verfälscht.

Vor dem Testen muß der Computer die Markenvereinbarungen in konkrete Operandenangaben umrechnen. Das macht er nach I ENTER. Bei fehlenden Zuordnungen (links stehende Marken) bricht er mit Angabe der vergeblich gesuchten Bezeichnung ab. Zu große Distanzen bei Relativsprüngen (Bereich von −128 bis 127) erkennt er jedoch nicht. Wenn das Adreßberechnen (Binden) vollständig klappt, entsteht wieder die normale Befehlsanzeige. Der Test kann beginnen.

Zum Testen gibt es vier Kommandos. **S** bewirkt die Ausführung des als ersten angezeigten Befehls. Danach gelangt (auch nach Sprüngen) der als nächstes anzuführende Befehl an die erste Stelle. Die aktuelle Flagbelegung erscheint rechts oben. Registerinhalte können mit **R** überprüft werden. Nach Rückkehr von DATA zu PROG (OFF) ist die Weiterführung des Tests uneingeschränkt möglich. **N** stoppt die Programmausführung erst bei Erreichen des als zweiten aufgelisteten Befehls. Damit lassen sich Unterprogrammsprünge und Schleifen beim Test übergehen. **H** vereinbart einen Haltepunkt, dessen Adresse vierstellig hexadezimal oder symbolisch (mit :) angefügt werden muß. Der nächste Stopp erfolgt dann erst bei Erreichen dieses Punktes im Programmablauf. **L** startet das Anwenderprogramm, ohne es wieder zu stoppen (Echtzeittest).

Das Betriebssystem benutzt die Register von %52 bis %7F. Sie dürfen vom Anwenderprogramm nicht verändert werden. Variationen des Anzeigebereichs mit **G** behindern das Testen nicht. So können Programmpassagen auch leicht übersprungen werden. Nach Verändern von Befehlen ist erneutes Binden nötig. Aber auch das verändert weder die rechts oben angezeigten Flags noch die Register außerhalb des Systembereichs von %52 bis %7F **(Abb. 45)**.

Wer nach längerer Übung das Programmieren in Maschinensprache beherrscht, kann sich mit dem Nutzen des Interruptsystems des Einchip-Mikrorechners befassen (vgl. JU+TE, 5/1989, ABC Einchip-Mikrorechner). Das Betriebssystem adressiert die Interrupt-Serviceroutinen mit Register-indirekten Sprüngen:

IVØ: %74 und %75
IV1: %76 und %77
IV2: %78 und %79
IV3: %7A und %7B
IV4: %7C und %7D
IV5: %7E und %7F

Das Betriebssystem benutzt den Timer TØ (IV4). Anwenderprogramme sollten ihn nicht in Anspruch nehmen, um die Bilderzeugung und Schritt-Tests nicht zu behindern.

Auch Programme für andere Rechner mit Einchip-Mikrorechner-Schaltkreis können wir auf unserem Computer entwickeln. Um sie auf den dort gültigen Adressen lauffähig zu machen, erfolgt das Binden mit I und der Adresse, auf der der oben angezeigte Befehl im anderen Rechner stehen soll. Anschließend kann z. B. per EPROM die entwickelte Software im Zielsystem installiert werden. Zuvor müssen wir allerdings unseren Computer mit einem EPROM-Programmierzusatz aufwerten.

TEIL II: AUSBAU ZUM ENTWICKLUNGSSYSTEM

Unser Grundgerät stellt bereits einen ernstzunehmenden Kleincomputer dar. Trotz extrem geringen gerätetechnischen Aufwands bietet er viele Möglichkeiten. Sie reichen vom Beschäftigen mit den Programmbeispielen bis zum Entwickeln eigener Software. Man kann sich sicherlich jahrelang intensiv mit unserem Computer beschäftigen und BASIC-Programme auch mit mehreren Kilobyte Länge entwickeln, ohne etwas zu vermissen. Es lassen sich sogar höhere Sprachen wie FORTH in das Programmsystem aufnehmen (implementieren), die mit der Standardperipherie völlig auskommen.

Wie viele andere Rechner läßt auch unser Computer den Anschluß weiterer Geräte zu. Grundsätzlich ist dabei keine EDV-Peripherie ausgenommen. Mit entsprechendem Aufwand können Massenspeicher (z. B. Diskettenlaufwerke), große Bildschirme, Drucker, Plotter (Zeichengeräte), Datenfernübertragungseinrichtungen u. a. m. angeschlossen werden. Dabei müssen wir aber die damit verbundene Effektivität bedenken. Sie hängt stark von einen standardisiertem Betriebssystem ab. In dieser Form, wie etwa CP/M (Mikroprozessor U 880 D), UNIX (Prozessor U 8001 D) oder MS-DOS (Prozessor 8086 und dessen Weiterentwicklungen) gibt es nichts für den Einchip-Mikrorechner. Der Ausbau in Richtung der üblichen Datenverarbeitungsanlagen erfordert einen Co-Prozessor, der die gerätetechnische Basis für ein solches Betriebssystem bietet. Insgesamt wird der Aufwand größer, als einen bereits in dieser Richtung entwickelten Computer (vgl. z. B. Kramer: Praktische Mikrocomputertechnik; Militärverlag der DDR, Berlin 1987) nachzubauen.

Der Einchip-Mikrorechner ist für dieses Anwendungsgebiet nicht ausgelegt. Seine Stärken zeigen sich beim Einsatz als Prozeßrechner. In der kommerziellen Anwendung sind das effektive Automatisierungsmittel, die mit wenig gerätetechnischem Aufwand viele Aufgaben der Meß-, Steuerungs- und Regelungstechnik erfüllen. Im Hobbybereich eignet sich unser Computer beispielsweise zum Steuern von Modelleisenbahnen und ähnlichen Spielzeugen mit Meß- und Stellmöglichkeiten. Er läßt sich auch zum Entwickeln von Programmen für spezielle Rechner mit dem Einchip-Mikrorechner U 882, U 884 oder anderen Typen mit dieser CPU verwenden (z. B. für Gewächshausregler, Heizungssteuerung).

Solche Entwicklungen arbeiten mit Maschinensprache, da sie die Fähigkeiten des Prozessors am besten nutzt und keine Zusatzprogramme wie den BASIC-Interpreter erfordert. Dabei ist ein Drucker wünschenswert, der mit Programmlisten die Übersicht besser wahren läßt. Das ist für die Fehlersuche auf diesem Sprachniveau sehr wichtig. Zum maschinenlesbaren Aufbewahren so entwickelter Programme eignen sich EPROM am besten. Sie können auf weiteren Modulen in unserem oder in anderen Computern gesteckt werden. Voraussetzung dafür ist aber ein EPROM-Programmierzusatz. Mit diesen beiden Geräten (Drucker und EPROM-Programmierzusatz) wird unser Computer zum Entwicklungssystem für Einchip-Mikrorechner-Programme. Die Peripherie des Entwicklungssystems erfordert jedoch eine universelle Ein-/Ausgabe-Schnittstelle.

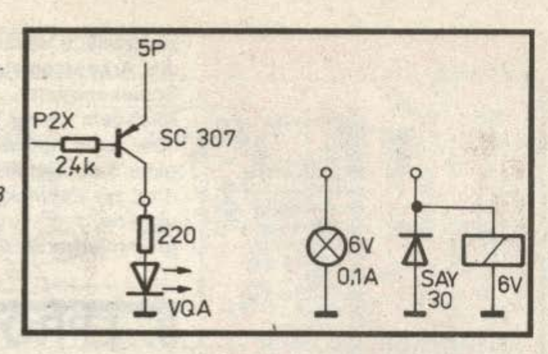
8. Ein-/Ausgabe-Schnittstelle

Wie wird die Prozeßperipherie angeschlossen?

Auf der Prozessorplatte des Computers sind die 16 Anschlüsse der Ports 2 und 3 (P2Ø bis P27 und P3Ø bis P37) zugänglich. Sie können für die Ein- und Ausgabe von Signalen im TTL-Pegel genutzt werden. Das Betriebssystem verwendet mit der Bildinterrupt-Serviceroutine P37 für Synchronsignale und P36 als akustischen Ausgang, das Magnetbandinterface zusätzlich P3Ø als Eingang. Der EPROM-Programmierzusatz und der Schreibmaschinenanschluß belegen weitere Signale, so daß fast nichts übrigbleibt. Solange die Entwicklungssystem-Peripherie nicht angeschlossen ist, stehen jedoch immerhin 13 Signale uneingeschränkt bereit.

Beim Port 3 liegen die Übertragungsrichtungen fest. P3Ø bis P33 sind Eingänge, P34 bis P37 Ausgänge. Die Signale P2Ø bis

P27 lassen sich in beiden Übertragungsrichtungen nutzen. Mit dem Laden des Registers Nr. %F6 wird festgelegt, welche Port-2-Signale Ausgänge sind. Eine Ø in diesem 8-bit-Register bewirkt die Ausgaberichtung auf der entsprechenden Bitposition von Port 2. Damit dabei auch 1-Pegel aktiv erzeugt wird und kein Open-drain-Betrieb entsteht, muß das Register Nr. %F7 den Wert 1 enthalten. Beide Register lassen sich in BASIC mit einer Prozedur laden. Mit
PROC SETRR [%F6, %7Ø1
werden z. B. P23 bis P27 Ausgänge. P2Ø bis P22 bleiben reine Eingänge. Sie sind über die Registeradresse 2 zugänglich. Mit
PROC SETR [2, Ausdruck]
gelangen die unteren acht Bit von „Ausdruck" in das Ausgaberegister von Port 2 und widerspiegeln sich an den in Ausgaberichtung vereinbarten Anschlüssen bis auf Widerruf mit den zugeordneten Signalpegeln (Ø:0...0,4 V; 1:2,4...5 V). Die Eingabe gelingt mit der Funktion GETR [2]. Dabei erhält man auf den unteren acht Bit die aktuelle Belegung der Signale P2Ø bis P27. Das ist bei Ausgabesignalen die zuletzt ausgegebene Information, bei den Eingabesignalen der von außen zugeführte Pegel. P3Ø bis P37 sind in gleicher Weise über die Registeradresse 3 zugänglich.
Bei der Beschaltung der Portanschlüsse ist zu beachten, daß nie Spannungen über 5P oder unter 00-Potential auftreten. In Eingaberichtung betriebene Einchiprechner-Anschlüsse stellen eine rein kapazitive Last (ca. 5 pF) dar, hier fließt praktisch kein Strom. Ausgänge können bei Ø-Pegel mit 2 mA, bei 1-Pegel mit 0,25 mA belastet werden. Ein Lastwiderstand gegen 00 muß daher mindestens 7,5 kΩ, gegen 5P mindestens 2,4 kΩ betragen. Größte Sorgfalt ist beim Umgang mit Steckverbindern, die zum Beispiel wahlweise den Anschluß des EPROM-Programmierzusatzes oder anderer Peripherie gestalten, geboten. **Grundsätzlich darf nur im ausgeschalteten Zustand gesteckt oder gezogen werden.** Beim Anschluß von Geräten mit eigener Stromversorgung schützt nur eine potentialtrennende Schnittstelle (vgl. Abb. 55) zuverlässig. Die Leuchtdioden in Abb. 46 können zu diesem Zweck z. B. von Sendedioden in MB-104-Schaltkreisen (Optokoppler) ersetzt werden. Wer diesen Aufwand scheut, sollte den möglichen Verlust des Einchiprechner-Schaltkreises und den daraus resultierenden Aufwand bedenken.



46 Einfacher Ausgabeverstärker für den Anschluß von LED, Glühlampen oder Relais an die Ports 2 und 3

Welche Prozesse lassen sich steuern?

Bevor wir zur Entwicklungssystem-Peripherie kommen, noch schnell ein Beispiel für die universelle Koppelbarkeit der E/A-Schnittstelle.
Das Anwendungsgebiet reicht von einfachen Lampensteuerungen bis zu solch komplizierten Aufgaben wie Modelleisenbahnen (Potentialtrennung nötig!). Soweit am Prozessor die genannten Bedingungen eingehalten werden (TTL-Pegel), sind keine Anwendungen ausgeschlossen. Reichen die Portsignale nicht aus, können Module mit Ein- oder Ausgabeschaltungen (z. B. DS 8282 D) statt Speicherschaltkreisen auf freien Modulplätzen als Erweiterung dienen. Sie sind dann wie Speicherzellen auf den entsprechenden Adressen zum Beispiel mit der Prozedur SETEB zugänglich.
Abb. 46 zeigt eine einfache Verstärkerschaltung, die den Anschluß von Leuchtdioden, kleinen Glühlampen oder Relais bis zur Leistungsgrenze der Stromversorgung gestattet. Beschalten wir damit die Port-2-Signale, läßt sich bereits ein einfaches, aber variantenreiches Lauflicht realisieren. Die programmtechnischen Möglichkeiten bieten diesbezüglichen Ideen breiten Spielraum. Unser Beispiel **(Abb. 47)** erzeugt eine pseudostochastische Binärfolge (BPSS) mit wählbarer Taktfrequenz.
Die Anweisung 2Ø vereinbart alle Port-2-Anschlüsse als Ausgang und gibt unserem Schieberegister (Variable A) einen von Ø verschiedenen Startwert. Es folgt die Eingabe der Taktperiode. Sinnvoll sind hier Zahlen zwischen 1 und 1ØØ. Ab der Anweisung 4Ø wird das Rückführen des Schieberegisters berechnet. Gut eignet sich hierzu die Antivalenz (XOR) der Bits 4 und 8. Die Variable C erhält diese beiden Stellen des Schieberegisters A, während alle anderen Positionen durch die AND-Verknüpfung mit Ø belegt werden. Von den vier möglichen Kombinationen der beiden inter-

47 BASIC-Programm Lauflicht

```
10  PRINT "LAUFLICHT"
20  PROC SETRR[%F6,1];
    LET A=1
30  INPUT "DAUER: "B
40  LET C=A$A%110
50  IF C=%100
    THEN LET A=A$O%8000
60  IF C=%10
    THEN LET A=A$O%8000
70  LET A=RL[A$A%80FF]
80  PROC SETR[2,A]
90  WAIT B
100 PROC SETR[%6D,0];
    CALL %C56
110 IF GETR[%6D]=0
    THEN GOTO 40
120 GOTO 30
```

```
Lauflicht

*LFL: 31 6Ø       Registerpointer
 EØØ2 E6 F6 ØØ    Port 2: Ausgabe
 EØØ5 E6 F7 Ø1    aktive pull-up
 EØØ8 D6 Ø8 DD    Bild löschen
 EØØB D6 22 E5    Byte-Eingabe
 EØØE 31 2Ø       Registerpointer
 EØ1Ø ØC FF       Schieberegister
 EØ12 1C FF       Anfangswert
*LF1: EØ EØ       C := Bit D8
 EØ16 76 E1 1Ø    D4 = 1 ?
 EØ19 6B :LF2     wenn nicht
 EØ1B EF          C := /C
*LF2: 1Ø E1       Schieben L
 EØ1E 1Ø EØ       Schieben H
 EØ2Ø 28 6C       Dauer
*LF3: 3A :LF3     Warteschleife
 EØ24 2A :LF3     Warteschleife
 EØ26 19 Ø2       Ausgabe
 EØ28 56 6D 8Ø    ASCII := 0
 EØ2B D6 ØC 56    Tastenabfrage
 EØ2E 76 6D 7F    neues ASCII ?
 EØ31 6B :LF1     wenn nicht
 EØ33 8B :LFL     Neustart
```

48 Maschinenprogramm Lauflicht

essierenden Bits erfüllen zwei die XOR-Verknüpfung. In diesen Fällen wird in der Zeile 5Ø oder 6Ø die Position 15 (ganz links), unseres Schieberegisters gesetzt.

Die Anweisung 7Ø besorgt das Schieben. Die AND-Verknüpfung löscht zunächst zur Vorbereitung des nächsten Zyklus die nicht benutzten Bits. Die Prozedur RL bringt die Information von Stelle 14 auf Stelle 15 usw. von Stelle 1 auf Stelle 2, von Ø auf 1 und auf Position Ø erscheint der ursprüngliche Inhalt von Bit 15 (XOR-Ergebnis). Die Ausgabe erfolgt anschließend mit der Prozedur SETR [2, A].

Nach der Warteanweisung folgt die dynamische Tastenabfrage. Das ist ein Betriebssystem-Unterprogramm, das mit dem Register Nr. %6D arbeitet. Solange keine Taste betätigt ist (in Register %6D bleibt der Inhalt Ø), setzt das Programm ab Zeile 4Ø seine pseudostoschastische Operation mit unveränderten Parametern fort. Sonst erwartet die Anweisung 3Ø eine neue Periodendauer-Eingabe.

Abb. 48 enthält ein Maschinenprogramm für die gleiche Funktion. Es benutzt das im Betriebssystem enthaltene Unterprogramm ab Adresse %22E5 für die Eingabe der Periodendauer als zweistellige Hexadezimalzahl und die Arbeitsregister Ø und 1 als Schieberegister.

Nachdem unser Computer nun über eine universelle Ein-/Ausgabe-Schnittstelle verfügt, zurück zur Entwicklungssystem-Peripherie, zuerst zum EPROM-Programmiergerät.

9. EPROM-Programmierzusatz

Wie müssen wir mit EPROM umgehen?

EPROM speichern Informationen in Form von kleinen Ladungen auf winzigen, völlig in Quarzglas (SiO_2) eingeschlossenen Siliziumstückchen. Die Ladungen können durch den normalen Betrieb im Rechner, bei dem nur Lese-Zugriffe möglich sind, nicht beeinflußt werden.

Beim Löschen erhalten die Ladungsträger per Bestrahlung mit UV-Licht genügend Energie, um das sonst ideal isolierende Quarzglas zu überwinden. Es lassen sich nur alle Speicherzellen gleichzeitig löschen, wobei alle Bits auf den Logikpegel 1 kommen. Dazu brauchen wir eine Quecksilber-Hochdrucklampe (HQL oder besser HQV), die zum Beispiel in Bestrahlungsgeräten (Höhensonne) eingesetzt werden. Sonnenlicht besitzt nicht genügend UV-Energie zum sicheren Löschen.

Beim Programmieren werden gezielt mit Hilfe einer hohen Spannung (Vpp je nach Typ 12,5 V, 21 V oder 25 V) Ladungsträger auf diejenigen Siliziumstückchen „geschossen", die beim späteren Lesen Ø-Pegel erzeugen sollen. Dazu muß die Programmierspannung etwa 50 ms lang auf die entsprechenden Speicherzellen einwirken. Da sich immer nur die 8 bit eines Bytes gleichzeitig behandeln lassen, benötigt diese Prozedur mindestens eine Minute je Kbyte.

Um EPROM an einem Ort programmieren und an anderem Ort einsetzen zu können, ist das Kontaktieren mit Steckfassungen üblich. Zum Vermeiden mechanischer Schäden müssen wir beim Stecken sehr vorsichtig sein. Befinden sich EPROM nicht in einer Fassung, droht außerdem die Gefahr der Zerstörung durch statische Ladungen (MOS-Schaltkreise!). Wir sollten daher grundsätzlich die Alu-Folientabletts oder leitende Schaumgummistückchen für den Transport von EPROM verwenden.

Erfordert das Programmieren spezielle Hardware?

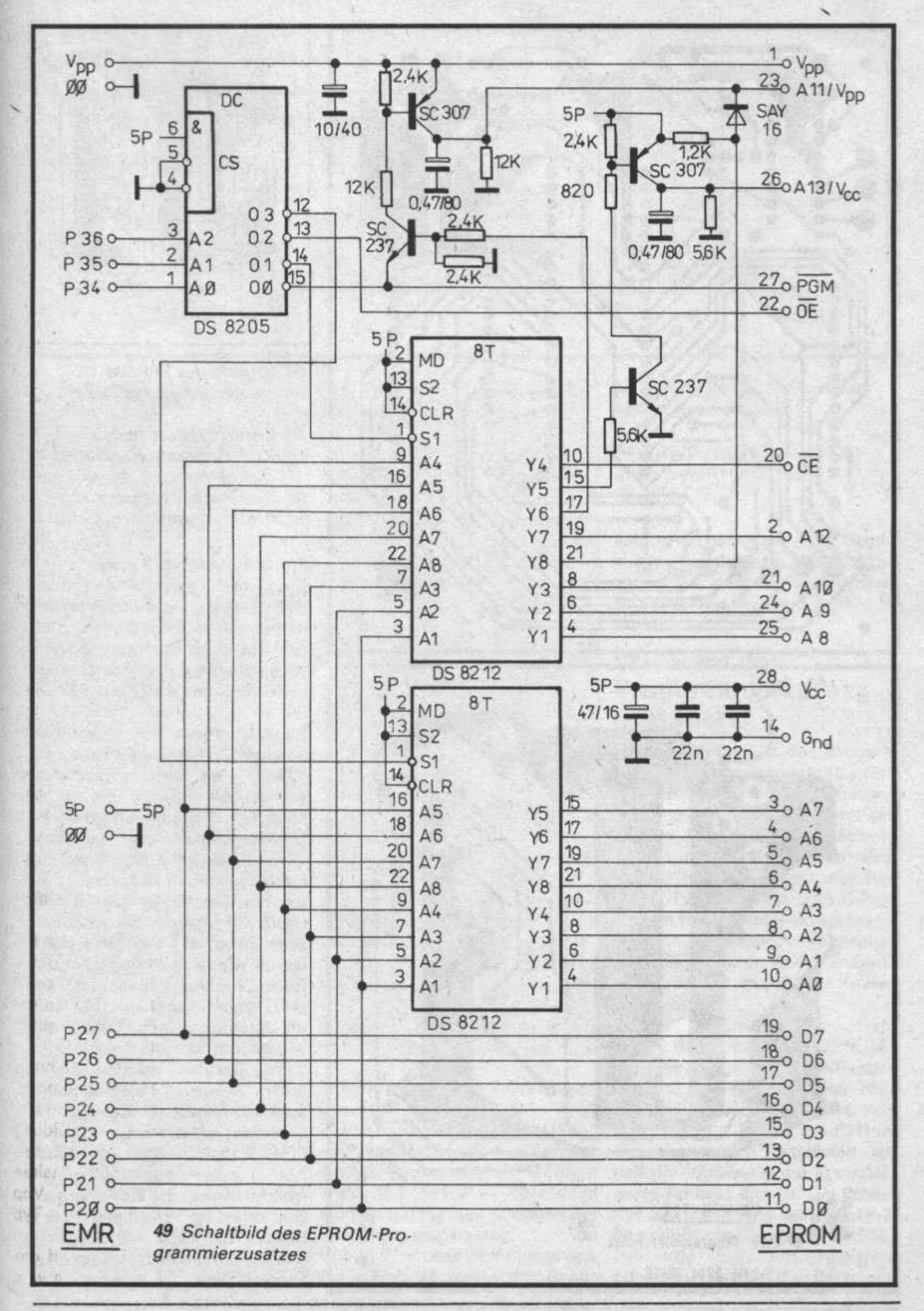
An den Steckplätzen, wo EPROM als Speicher eines Mikrorechners benutzt werden, sind nur die für das Lesen nötigen Spannungen und Signale üblich. Deshalb braucht unser Computer eine zusätzliche Baugruppe zum Programmieren von EPROM. **Abb. 49** zeigt deren Schaltbild. Sie eignet sich zum Programmieren und Lesen der Typen U 2716 C, U 2764 C und U 27128 C. Die Anschlußbezeichnung bezieht sich auf die beiden letztgenannten 28-poligen Schaltkreise.

Die Steuersignale P34, P35 und P36 (Port-3-Ausgänge des Einchip-Mikrorechners, siehe auch Abb. 2) bestimmen mittels des DS 8205 D den Zustand der Schaltung. Solange P36 1-Pegel besitzt, bleibt alles passiv. Sonst gilt die folgende Zuordnung:

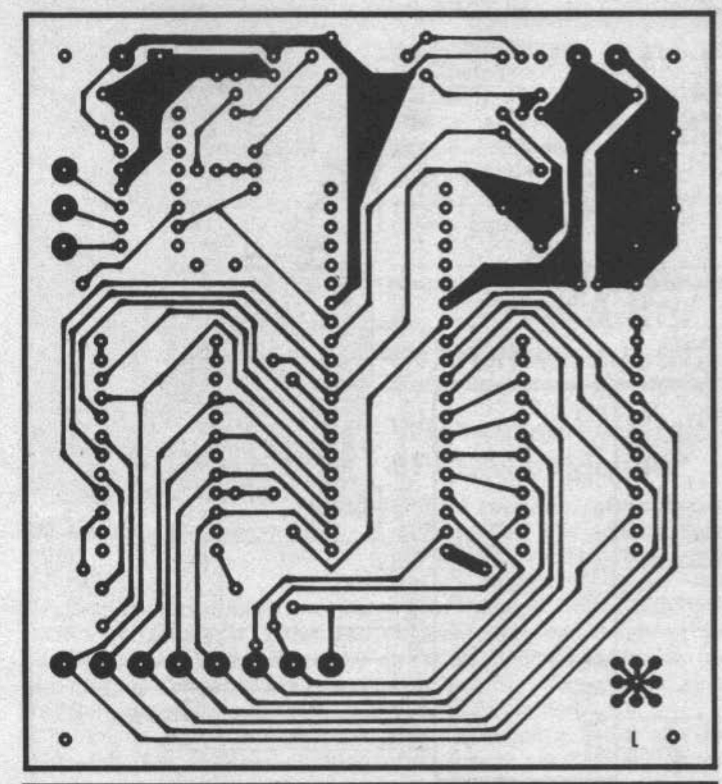
Mit P34=1 und P35=1 werden die unteren acht Adreßbit (AØ bis A7) für den EPROM vom Port 2 (Signale P2Ø bis P27) in den unten (Abb. 49) dargestellten DS 8212 D geladen.

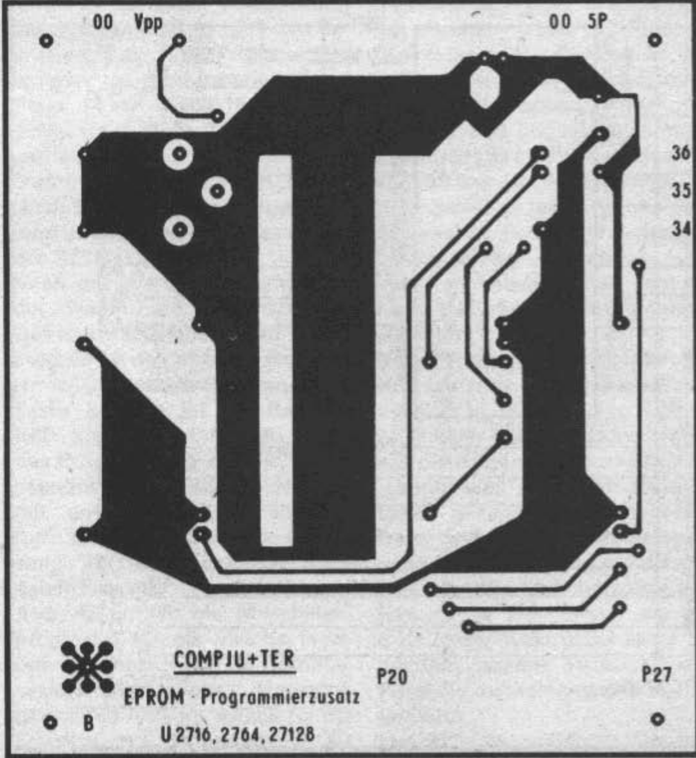
Bei P34=1 und P35=Ø speichert das andere DS 8212 D die höheren Adreßbits und zwei Steuersignale. Sie betreffen die Bildung der Schaltkreisauswahl $\overline{CE}$ des EPROM und das Durchschalten der Programmierspannung Vpp auf Pin 23, was nur der Typ U 2716 C erfordert.

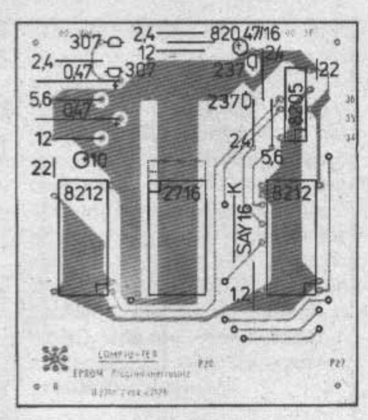
Mit P34=Ø und P35=1 wird das Steuersignal $\overline{OE}$ zum Lesen vom



***49** Schaltbild des EPROM-Programmierzusatzes*





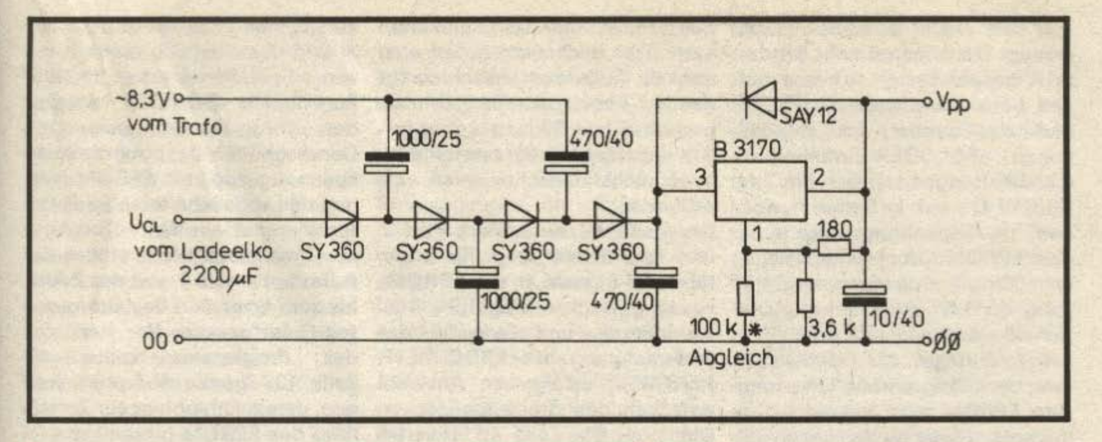


50 Lötseite des EPROM-Programmierzusatzes (links)

51 Bestückungsseite des EPROM-Programmierzusatzes

52 Bestückungsplan des EPROM-Programmierzusatzes

EPROM über Port 2 aktiv.
Die Kombination P34=Ø und P35=Ø ist während des Programmierimpulses erforderlich. Derweil stellt der Computer die einzuspeichernde Bitkombination über die Signale P2Ø bis P27 bereit.
Beim Übergang von einem zum anderen Zustand muß stets mit P36=1 alles passiv geschaltet werden, um Port 2 in die für die nächste Aktion erforderliche Übertragungsrichtung initialisieren und die ggf. nötige Bitkombination ausgeben zu können.
Bei der Gestaltung der Lötseite **(Abb. 50)** und der Bestückungsseite **(Abb. 51)** der Leiterplatte wurde wieder auf möglichst einfache Topologie geachtet. Die zehn freien Durchkontakte sind im Bestückungsplan **(Abb. 52)** wieder mit kleinen Kreisen markiert. Darüber hinaus dienen auch viele Bauelementeanschlüsse dem Verbinden von Löt- und Bestückungsseite. Für den EPROM kann eine 28- oder 40-polige Schwenkhebelfassung verwendet werden. Eine normale Steckfassung verschleißt in der Regel nach etwa 100 Steckvorgängen. Ungenutzt bleiben beim Stecken stets nur über dem



53 Schaltbild der Programmierspannungserzeugung

```
10  PROC SETR[%FB,%10];
    PROC SETR[%F7,1];
    PROC SETR[3,%70];
    PROC SETR[%F6,0];
    LET A=0;
    GOSUB 500
20  CALL %8DD;
    PRINT "U2716-HANDLER ";
    PRINT "1=HEX-EINGABE";
    PRINT "2=BRENNEN"
30  PRINT "3=LESEN";
    PROC SETR[%FB,%90];
    INPUT "AUSWAHL:"E;
    IF E>3
    THEN GOTO 10
40  INPUT "RAM-ANF:"C;
    INPUT "RAM-END:"D;
    LET A=C
50  IF E>1
    THEN GOTO 90
60  PTH A,":"GETEW[A];
    INPUT " NEU :"B;
    PROC SETEW[A,B]
70  LET A=A+2;
    IF A>D
    THEN GOTO 10
80  GOTO 60
90  INPUT "ROM-ANF:"F;
    LET A=F,G=D-C+F+1,I=C;
    IF G>%800
    THEN GOTO 10
100 IF E=3
    THEN GOTO 240
110 PRINT "VPP PRUEFEN !";
    LET B=GTC;
    PROC SETR[%FB,%10]
120 GOSUB 550;
    PROC SETR[3,%40];
    PROC SETR[2,GETEB[I]]
130 PROC SETR[3,0];
    WAIT 45;
    PROC SETR[3,%40]
140 LET A=A+1,I=I+1;
    IF A<G
    THEN GOTO 120
150 LET A=F,G=0,H=0,J=-1
160 GOSUB 600
170 IF B<>GETEB[C]
    THEN LET G=G+1
180 IF B<>%FF
    THEN LET H=H+1
190 CALL %E3C0
200 LET A=A+1,C=C+1;
    IF C<=D
    THEN GOTO 160
210 PROC SETR[3,%70];
    IF H=0
    THEN PRINT "ROM LEER"
220 ELSE ;
    PRINT G," FEHLER"
230 PTH "CRC:"J;
    PROC SETR[%FB,%90];
    LET B=GTC;
    GOTO 10
240 PROC SETR[%FB,%10]
250 GOSUB 600;
    PROC SETEB[I,B];
    LET A=A+1,I=I+1;
    IF A<G
    THEN GOTO 250
260 GOTO 150
500 PROC SETR[3,%50];
    PROC SETR[2,A/256$0%48]
510 PROC SETR[3,%10];
    PROC SETR[3,%F0]
520 PROC SETR[2,A];
    PROC SETR[3,%30]
530 PROC SETR[3,%70];
    RETURN
550 PROC SETR[3,%50];
    PROC SETR[2,A/256$0%E8];
    GOTO 510
600 GOSUB 500;
    PROC SETR[%F6,%FF];
    PROC SETR[3,%60];
    PROC SETR[3,%20]
610 LET B=GETR[2];
    PROC SETR[3,%60];
    PROC SETR[%F6,0];
    RETURN

Maschinen-Unterprogramm:

E3C0: 31 30 48 23 B2 42 28 E4
E3C8: F0 E4 56 E4 0F B2 42 28
E3D0: E4 F0 E4 90 E4 58 E4 56
E3D8: E4 1F B2 43 38 E4 48 E5
E3E0: E0 E4 56 E4 F0 B2 43 56
E3E8: E5 E0 B2 52 28 E4 38 E5
E3F0: AF 0D 00 00 00 00 00 00
```

54 BASIC-Programm zum Programmieren von EPROM (U 2716 C)

EPROM befindliche Anschlüsse. Die Spannungsversorgung (5 V/ 200 mA und Vpp/30 mA) kann über das Netzteil des Computers erfolgen, wenn dort ein kräftiger Trafo (8,3 V/1,3 A) verwendet worden ist. Für das Erzeugen der Programmierspannung von 12,5 V, 21 V oder 25 V eignet sich die in **Abb. 53** dargestellte Schaltung. Sie basiert auf einer Spannungsvervielfachung. Der Ausgangswert muß mit der Widerstandskombination am B 3170 V abgeglichen werden.

Wie wird der Programmierzusatz gesteuert?

Am effektivsten ist das Steuern des Programmierzusatzes mit Maschinen- bzw. Assemblerprogrammen. Höhere Sprachen erfordern merklich mehr Rechenzeit. Für das Handhaben des Typs U 2716 C (sowjetischer Typ K 573 PФ 2) ist ein BASIC-Programm jedoch noch akzeptabel. Unser Programmbeispiel (Abb. 54) realisiert das Standardverfahren (50 ms Programmierzeit je byte).

Das Unterprogramm ab **Zeile 500** lädt die EPROM-Adresse in die beiden Auffangsregister (DS 8212 D) des Programmierzusatzes. Es setzt voraus, daß Port 2 (P20 bis P27) in Ausgaberichtung initialisiert ist und die Variable A die einzustellende Adresse enthält. Die Prozedur SETR mit Registeradresse 3 beeinflußt die Steuersignale P34, P35 und P36, mit Registeradresse 2 jedoch die Belegung

von Port 2. Der Ausdruck A/256 erzeugt die höheren acht Bit der in A gespeicherten Adresse auf den bei der Ausgabe mit SETR benutzten unteren acht Bitpositionen. Die ODER-Verknüpfung mit %48 bringt zusätzlich die Bits D6 und D3 auf 1. Dadurch werden die Spannungsversorgung des EPROM (Vcc) und 1-Pegel am Programmierspannungseingang (A11/Vpp) gesichert. Die Schaltkreisauswahl $\overline{CE}$ erhält den aktiven Ø-Pegel, das Durchschalten der Programmierspannung zum EPROM wird gesperrt. Damit entsteht die für das Lesen nötige Steuersignalbelegung.

Die Anweisung 51Ø überträgt die so zusammengestellte Bitkombination in den in Abb. 49 oben dargestellten DS 8212 D. Die Zeile 52Ø lädt das andere Auffangregister mit den niederen acht Adreßbits, bevor die Anweisung 53Ø die Steuerung mit P36=1 passiviert und den Rücksprung (RETURN) in das Hauptprogramm auslöst.

Das Unterprogramm ab **Zeile 55Ø** dient auch dem Einstellen der Adresse aus der Variablen A, nur daß hier $\overline{CE}$ den passiven 1-Pegel erhält und das Durchschalten der Programmierspannung vorbereitet wird. Damit entsteht die für das Programmieren nötige Signalbelegung.

Das dritte Unterprogramm ab **Zeile 6ØØ** realisiert das Lesen der EPROM-Zelle, deren Adresse die Variable A bereitstellt, in die Variable B. Dazu wird nach dem Laden der Adresse (GOSUB 5ØØ) Port 2 mit SETR[%F6,%FF] in Eingaberichtung initialisiert. Nach Aktivieren des Signals $\overline{OE}$ erhält die Variable B über P2Ø bis P27 die Information vom EPROM. Es folgen das Passivieren von $\overline{OE}$, das Initialisieren von Port 2 in Ausgaberichtung und der Rücksprung.

Das **Hauptprogramm** blockiert während jedes Datenaustausches mit dem EPROM-Programmierzusatz durch PROC SETR [%FB,%1Ø] die Interrupt-Annahme. Das ist nötig, damit die Bilderzeugung, die P36 beeinflußt, nicht die Kommunikation stört. Der Bildschirm bleibt also dunkel. Außerdem beschleunigt das die Arbeit des BASIC-Interpreters. Ohne Bilderzeugung lassen sich diese Programmteile jedoch nicht stückchenweise ausprobieren.

Die Zeile 1Ø initialisiert Port 2 und Port 3 und sorgt für ungefährliche Signale an der EPROM-Fassung. Nach Anzeige des Auswahl-Menüs und Freigabe der Bilderzeugung mit PROC SETR [%FB,%9Ø] erfolgt die Auswahl zwischen den drei angebotenen Diensten. Die Zeile 4Ø erwartet dann das Eingeben von RAM-Anfangs- und -Endadresse in die Variablen C und D. Damit wird der Arbeitsspeicherbereich im Computer festgelegt. Um einen Fehler bei der RAM-Adreßeingabe zu korrigieren, muß neu begonnen werden.

Die **Auswahl 1** führt zur RAM-Eingabe, die hexadezimal erfolgen sollte. Sie behandelt immer gleich zwei Bytes auf einmal. Das erfordert eine geradzahlige RAM-Anfangsadresse. Nach Anzeige des alten RAM-Inhalts fordert die Zeile 6Ø die Eingabe des neuen Inhalts. In Zeile 7Ø erfolgen das Weiterstellen der Adresse A und bei Erreichen des Endes des vereinbarten Speicherbereichs das Anbieten einer neuen Auswahl zum Übertragen des Eingebenen in den EPROM (Programmieren).

Bei Auswahl 2 oder 3 wird mit Anweisung 9Ø die EPROM-Anfangsadresse in die Variable F eingegeben. A erhält diese Zahl als Startwert, I die RAM-Anfangsadresse. In G wird die Grenze errechnet, bis zu der (ausschließlich) der EPROM zu behandeln ist. Übersteigt G den Wert %8ØØ, reicht die Kapazität des U 2716 C nicht aus.

Bei **Auswahl 2** fordert unser Programm mit der Anweisung 11Ø das Überprüfen der Programmierspannung Vpp. Das Bestätigen erfolgt mit einer beliebigen Taste. Nach Abschalten der Bilderzeugung beginnt das Programmieren (Brennen). Den Zyklus eröffnet die Zeile 12Ø mit dem Einstellen der EPROM-Adresse aus A und der Ausgabe eines Bytes von Speicheradresse I über Port 2. Die Zeile 13Ø realisiert den Programmierimpuls mit Durchschalten der Programmierspannung Vpp zum EPROM. Wegen des abgeschalteten Bildinterrupts dauert ein WAIT-Durchlauf nur 1 ms. Nach Weiterstellen der Adressen A und I wird der Zyklus bis zum Erreichen des Grenzwertes G fortgesetzt. Vor Benutzen des Programms sollte die Zeile 13Ø genau überprüft werden, da ein Fehler hier zur Zerstörung des EPROM führen kann.

Mit der Zeile 15Ø beginnt das Prüfen. G zählt die Fehler, H die von %FF verschiedenen EPROM-Inhalte. In J wird die zyklische redundante Kontrollsumme **CRC** gebildet. Es ist üblich, diese Zahl auf programmierten EPROM zu vermerken, um sich später mit wiederholter CRC-Berechnung vom fehlerfreien Speicherinhalt überzeugen zu können. Dieser Algorithmus läßt sich in BASIC sehr schlecht realisieren, daher ruft die Anweisung 19Ø ein entsprechendes Maschinenprogramm.

Die Zeile 21Ø erzeugt eine ungefährliche Belegung von P34, P35 und P36 und ermittelt ggf. einen leeren EPROM dadurch, daß alle untersuchten Bytes im gelöschten Zustand vorgefunden wurden. Andernfalls bringt die Zeile 22Ø die Anzahl der Fehler, die im Normalfall gleich null ist, zur Anzeige. Danach folgen noch die Angabe der Kontrollsumme CRC, das Freigeben der Bilderzeugung und das Abwarten einer beliebigen Tastenbetätigung.

Bei **Auswahl 3** führt das Programm ab Zeile 25Ø das Übertragen des EPROM-Inhaltes in den RAM des Computers aus. Anschließend folgt auch hier der beschriebene Prüfalgorithmus.

Wie geben wir das Steuerprogramm ein?

Mit dem BASIC-Editor werden zuerst alle Anweisungen eingege-

ben. Das Programm belegt den Bereich von %EØØØ bis %E3BF. Die letzten beiden Bytes lauten %52 und %ØD. Das Maschinenprogramm geben wir am besten mit der RAM- und Registeranzeige ein. Mit GE3BE suchen wir das Ende des BASIC-Programms, das wegen eines Leerzeichens mehr oder weniger auch einen abweichenden Speicherbedarf haben kann. Da 13 byte Reserve bestehen, können wir uns den Gegebenheiten anpassen.
In Anschluß an das BASIC-Programm folgt die Eingabe des Maschinenprogramms bis %ØD und %ØØ. Damit wird es für das Betriebssystem ein Teil des BASIC-Textes. LIST kommt mit dem Maschinencode allerdings nicht klar, was jedoch nicht schadet. Wenn das erste Byte (%31) nicht auf Adresse %E3CØ steht, muß anschließend noch die BASIC-Anweisung 19Ø korrigiert werden. Dann ist es endlich fertig und kann von BASIC aus mit dem Kommando SAVE auf Kassette gespeichert werden.

Was ist beim Anwenden zu beachten?

Das Programm gestattet die beliebige Auswahl des Arbeitsspeicherbereiches (RAM). Es läßt sich dadurch an jede konkrete Ausstattung des Computers anpassen. Bei minimaler RAM-Kapazität belegt das Programm bereits das einzige Anwender-Kilo. Als Arbeitsbereich bleibt das erste Viertel des Systemkilos (%FCØØ bis %FCFF). Hiermit kann ein U 2716 C Record für Record in acht Schritten behandelt werden.
Bei der ersten Inbetriebnahme sollte bei abgeschalteter Programmierspannung Vpp mit dem Einlesen eines Records begonnen werden. Stimmt der danach mit der RAM- und Registeranzeige lesbare Datensatz nicht mit dem betreffenden EPROM-Inhalt (der natürlich bekannt sein muß) überein, sind wieder falsche Verbindungen und Leiterzugunterbrechungen zu suchen. Das Unterprogramm ab Zeile 5ØØ variiert auch P36 und P37, so daß sich der Ablauf sowohl optisch als auch akustisch gut verfolgen läßt. An den Adreßeingängen AØ bis A7 des EPROM müssen Schwingungen mit von Bit zu Bit halbierter Frequenz beim Lesen und Prüfen auftreten. Alle Ausgangssignale des anderen DS 8212 D bleiben für die Dauer eines Records unverändert. A8 bis A1Ø führen dessen Adresse, A11/Vpp und A13/Vcc 1-Pegel und A12, $\overline{CE}$ und Anschluß 17 des DS 8212 D Ø-Pegel. Am DS 8205 D müssen sich die Ausgänge an den Pins 12, 13 und 14 ständig verändern, während Pin 15 konstant 1-Pegel erzeugt. Erst wenn das Lesen hundertprozentig klappt, können wir uns ans Programmieren wagen.
Die Programmierspannung Vpp (25 V) darf ständig am Programmierzusatz angeschlossen sein. Ein EPROM kann aber erst gesteckt werden, wenn das steuernde Programm läuft und eine Eingabe erwartet. Eine falsche Polung birgt die Gefahr der Zerstörung des EPROM! Bevor mit RESET ein Abbruch des Programmierprogramms erfolgt, ist der EPROM wieder aus der Fassung zu nehmen.
Vor dem Programmieren muß der EPROM leer sein. Kontrollieren läßt sich das mit Auswahl 3, wobei der RAM-Bereich von Ø bis %7FF gewählt werden kann. Da sich dort kein RAM befindet, ändert sich nichts am Speicherinhalt des Computers. Der Test im Anschluß an dieses (sinnlose) Lesen ermittelt, ob der EPROM leer ist. Das dauert insgesamt knapp dreieinhalb Minuten.
Nach der Eingabe des gewünschten EPROM-Inhalts in den Arbeitsbereich (z. B. %FCØØ bis %FCFF) mit Auswahl 1 oder mit der RAM- und Registeranzeige DATA kann mit Auswahl 2 programmiert werden. Das dauert für einen Record (256 bytes) knapp 40 Sekunden. Der Test einschließlich der CRC-Berechnung bezieht sich auf den jeweiligen Bereich. Zum Ermitteln der gesamten Kontrollsumme eignet sich das Lesen auf einen nicht mit Speicherschaltkreisen bestückten 2K-byte-Bereich (z. B. Ø bis %7FF) mit Auswahl 3.
Im Computer gespeicherte Daten können direkt in einen U 2716 C übertragen werden. Zum Programmieren von Betriebssystem-EPROM eignen sich die Arbeitsbereiche %Ø8ØØ bis %ØFFF (EPROM 1) und %2ØØØ bis %27FF (EPROM 2). Das hier in Wirklichkeit kein RAM im Computer bestückt ist und statt dessen aus dem Betriebssystem gelesen wird, stört nicht. Das vollständige Programmieren eines U 2716 C dauert gut fünf Minuten.

Funktioniert das auch mit anderen EPROM-Typen?

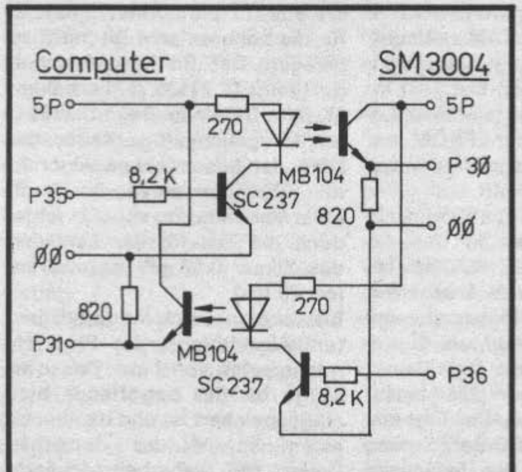
Das beschriebene Programm ist gedacht für U 2716 C und kompatible Schaltkreise der verschiedensten Hersteller. Für die anderen Typen (U 2764 C und U 27128 C) eignet sich diese Software nicht. Hier muß ein anderes Signalspiel realisiert werden.
Das Lesen gelingt wie gewohnt mit $\overline{CE}$=Ø und $\overline{OE}$=Ø. Zum Programmieren muß ebenfalls $\overline{CE}$=Ø (im Gegensatz zum U 2716 C) realisiert werden, wobei das Durchschalten von Vpp auf Pin 23 unterbleibt. Dazu sind die Bits D5 und D7 im DS 8212 D für die höheren acht Bit mit Ø zu belegen. Das Bit D6 unterscheidet beim U 27128 C die beiden 8K-Byte-Bereiche. Beim U 2764 C ist es gleichgültig. Ansonsten kann der beschriebene Algorithmus übernommen werden. Nach jeder Veränderung von A11 ist jedoch die Zeit für das Umladen des Elkos (0,47 µF) abzuwarten (ca. 20 ms).
Besser eignet sich ein „intelligenter" Algorithmus, der Programmierimpulse von 1 ms Dauer erzeugt, bis das betreffende byte eingespeichert ist und dann noch einen Impuls der doppelten Dauer zur Sicherheit anhängt.

Dabei werden meist entschieden weniger als 50 ms je byte benötigt, so daß sich kürzere Programmierzeiten ergeben. Dieser Algorithmus läßt sich nicht in BASIC realisieren.
Bei den höher integrierten Typen unterscheiden sich die Eigenschaften zwischen Schaltkreisen verschiedener Hersteller teilweise erheblich. Auch mit dem Standard-Algorithmus ist man vor schmerzlichen Überraschungen nicht sicher. Zu beachten sind die Programmierspannungen von 21 V bzw. 12,5 V (bei den neueren Typen 2764 A und 27128 A), deren Überschreitung den EPROM zerstört. Für den U 2732 C eignet sich die Hardware nicht. Hier müßte $\overline{OE}$ statt A11 mit dem Programmierimpuls gemultiplext werden.

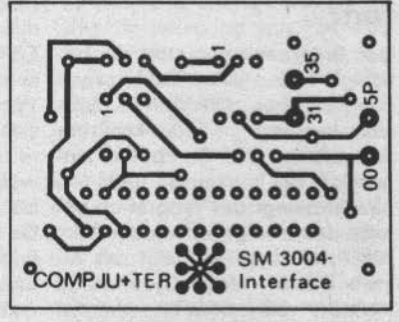
10. Druckeranschluß

Womit können wir drucken?

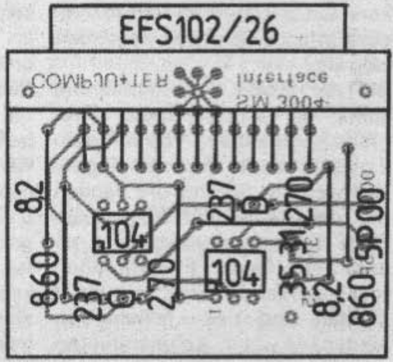
Drucker, die speziell für den Anschluß an Computer gefertigt werden, sind schwer zu beschaffen und außerdem vergleichweise teuer. Für unsere Zwecke erweist sich die Schreibmaschine „Erika 3004 electronic" als wesentlich effektiver. Sie besteht aus einem Typenrad-Druckwerk, einer angenehm gestalteten Elastomer-Tastatur und einer elektronischen Steuerung, die als Kern den Einchip-Mikrorechner UB 8840 M enthält. Unabhängig vom Computer-Anschluß ist dieses Gerät eine hochwertige Schreibmaschine, mit der besonders bei Karbon-Farbband sehr schöne und fehlerfreie Schriftbilder erzeugt werden können. Für unsere Zwecke empfiehlt sich aus Kostengründen jedoch mehr der Einsatz von textilem Farbband.
Für den Datenaustausch über den Interface-Steckverbinder verwendet die Schreibmaschine 3004 das Serieninterface (SIO) des Einchip-Mikrorechners. Daher müssen Zeichencodes und Druck-Kommandos im seriellen Format angeboten werden. Zur Geschwindigkeitsanpassung erzeugt die Schreibmaschine ein Rückmelde-Signal, das mit 1-Pegel den Empfang und mit Ø-Pegel die Ausführung eines Zeichens oder Kommandos quittiert. Für diese Signale werden die Pins P3Ø und P36 des in der Schreibmaschine enthaltenen Einchip-Mikrorechners verwendet.
Sie liegen direkt am Interface-Steckverbinder an und sind daher sehr vorsichtig zu behandeln. Um allen denkbaren Schäden, für die der Hersteller natürlich nicht haften würde, vorzubeugen, empfiehlt sich dringend eine potentialtrennende Interface-Schaltung **(Abb. 55). Abb. 56** und **Abb. 57** zeigen Leiterzugbild und Bestückungsplan einer geeigneten Schnittstelle. Die Leiterplatte sitzt am 26poligen Flachstecker (EFS), der in die Interface-Buchse der Schreibmaschine paßt. Deren Konstruktion läßt wenig Platz, so daß die Schrauben zur Befestigung des Steckverbinders in der Leiterplatte versenkt werden müssen. Außerdem empfiehlt es sich, die 14 der Schreibmaschine nächsten Anschlußstifte zu kürzen und möglichst flach zu verlöten.
Für die Zeichendarstellung gilt nicht der Standardcode (ASCII). Statt dessen ist die Position des gewünschten Zeichens auf dem Typenrad anzugeben. Mit Hilfe einer entsprechenden Tabelle für



55 Schaltbild des potentialtrennenden SM 3004-Interface



56 Leiterzugbild des SM 3004-Interface



57 Bestükkungsplan des SM 3004-Interface

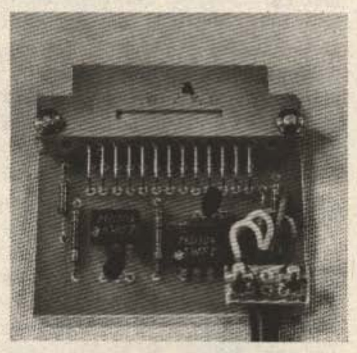
	Start bit	DØ	D1	D2	D3	D4	D5	D6	D7	Stoppbits		
Zähler	11	10	9	8	7	6	5	4	3	2	1	0

58 Serielles Datenformat

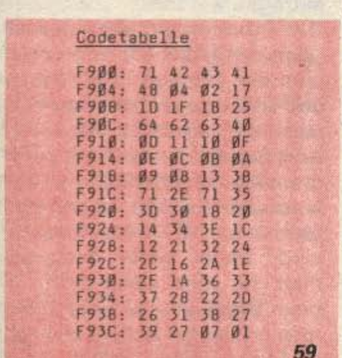
die im Computer verwendeten ASCII läßt sich dieser Code jedoch leicht erzeugen. Zusätzlich gibt es einige Druck-Komandos, von denen das Leerzeichen (%71) und die Zeilenschaltung (%77) die wichtigsten sind.

Wie steuert der Computer die Schreibmaschine?

Die Übertragung gelingt am einfachsten mit dem seriellen Interface. Es benutzt P37 als Ausgang. Da dieses Signal in unserem Computer die Synchronimpulse zur Bilderzeugung ausgibt, würden sich Druckersteuerung und Bilderzeugung gegenseitig ausschließen. Die Übertragungsrate von nur 1200 bit/s läßt aber genügend Zeit, das serielle Format **(Abb. 58)** programmtechnisch zu realisieren und P35 für die Ausgabe zu verwenden. Die dafür nötigen Programme lassen sich im Speicherbereich von %F9ØØ bis %FA84 unterbringen. Dabei bleibt genug Platz für die Markentabelle, deren Beginn (höchste Adresse) mit %FBFF festgelegt wird. Die Eingabe der Codetabelle **(Abb. 59)** erfolgt mit der RAM- und Registeranzeige DATA, die der Programme mit PROG. Voraussetzung ist natürlich die Belegung dieses Bereichs mit einem RAM-Schaltkreis. Beim Nutzen anderer Adressen müssen die ersten beiden Befehle des Druck-Unterprogramms **(Abb. 60)** an die jeweilige Lage der Codetabelle angepaßt werden.



Interface zum Anschluß der Schreibmaschine „Erika 3004 electronic"



Codetabelle

F9ØØ:	71	42	43	41
F9Ø4:	48	Ø4	Ø2	17
F9Ø8:	1D	1F	18	25
F9ØC:	64	62	63	4Ø
F91Ø:	ØD	11	1Ø	ØF
F914:	ØE	ØC	ØB	ØA
F918:	Ø9	Ø8	13	3B
F91C:	71	2E	71	35
F92Ø:	3D	3Ø	18	2Ø
F924:	14	34	3E	1C
F928:	12	21	32	24
F92C:	2C	16	2A	1E
F93Ø:	2F	1A	36	33
F934:	37	28	22	2D
F938:	26	31	38	27
F93C:	39	27	Ø7	Ø1

59

UP Zeichen drucken

```
*DRU: ØC F9      Codetabelle H
 F942 26 E1 2Ø   Codetabelle L
 F945 82 1Ø      SM-Code
*DR1: 76 Ø3 Ø2   SM bereit ?
 F94A EB :DR1    sonst warten
 F94C 9A :DR1    Entstören
 F94E 8F         Interruptsperre
 F94F 8C ØB      Bitzähler := 11
*DR5: 56 Ø3 DF   P35 := 0
*DR6: 9C B6      Zeitkonstante
*DR2: FF         Warteschleife
 F957 9A :DR2    819 µs
 F959 ØØ E8      Bitzähler -1
 F95B 6B :DR3    wenn fertig
 F95D A6 E8 Ø3   noch Daten ?
 F96Ø 7B :DR4    wenn nicht
 F962 BØ FA      Anmeldungen =0
 F964 EØ E1      Schieben rechts
 F966 FB :DR5    wenn 0-Bit
*DR4: 46 Ø3 2Ø   P35 := 1
 F96B 8B :DR6    zur Zeitschleife
*DR3: 9F         Int.-Freigabe
 F96E 76 FA Ø4   Quittung ?
 F971 6B :DR3    wenn nicht
 F973 AF         Rücksprung
```

60

Das Unterprogramm DRU ermittelt aus der Codetabelle, die die Schreibmaschinen-Codes in der Reihenfolge der ASCII enthält, im Arbeitsregister 1 das auszugebene Byte. An der Marke DR1 wird die Bereitschaft der Schreibmaschine abgewartet. Zum Übergehen von Störimpulsen erfolgt dieser Test gleich 256mal. Angesichts der relativ langsamen Verarbeitung seitens der Schreibmaschine können wir uns solchen Luxus ohne Zeitverlust leisten. Damit die Bildinterrupts nicht das Zeitregime stören, werden sie einfach verboten. Dadurch zuckt das Bild bei der Ausgabe jedes Zeichens. Das Arbeitsregister 8 zählt die Bits des seriellen Datenformats. Nach Einstellen des Startwerts 11 beginnt mit P35:=Ø das Startbit.

Beim Bemessen der Warteschleife wurde die Programmlaufzeit zum Ermitteln des nächsten Ausgabepegels berücksichtigt, so daß die nötige Bitzeit von 833 µs genau genug realisiert wird. Solange der Bitzähler noch mindestens auf drei steht, kommen die einzelnen Bits des Arbeitsregisters Ø zur Ausgabe über P35. Vor dem letzten Datenbit ist das Löschen der Interruptanmeldungen nötig. Da es auch vorher ohne negative Folgen ausgeführt werden darf, erfolgt es der Einfachheit halber vor der Ausgabe jedes Datenbits. Je nach der über das C-Flag ausgewerteten Bit-Belegung beginnt die nächste Warteschleife mit 1-Pegel (über DR4) oder Ø-Pegel (über DR5) an P35.

Der Ausgabe der acht Datenbits folgen zwei Bitzeiten mit 1-Pegel als Stoppbits. Danach wird beim Bitzählerstand Ø der Zyklus über DR3 verlassen. Dort folgen die Freigabe der Bilderzeugung und das Warten auf die Quittung der Schreibmaschine, die anhand der Interruptanmeldung von P31 im Bit D2 des Anmelderegisters mit der Adresse %FA leicht zu testen ist. Der Rücksprung beendet dieses Unterprogramm zum

UP Drucken des Bildinhalts

```
*HCB: 6C FD      BWS-Adresse H
 F976 7C 8Ø      Bildende L
*HC1: ØØ E7      Adresse -1
 F97A 82 16      ASCII
 F97C A6 E1 2Ø   Leerzeichen ?
 F97F 6B :HC1    wenn nicht
 F981 58 E7      letzte Adr.
 F983 7C ØØ      erste Adresse
*HC2: 82 16      ASCII
 F987 D6 :DRU    Drucken
 F98A D6 :NBA    Adresse +1
 F98D A2 57      fertig ?
 F98F FB :HC2    wenn nicht
 F991 AF         Rücksprung
```

61

UP Erhöhen der BWS-Adresse

```
*NBA: 7E         Adresse +1
 F993 66 E7 ØD   Zeilenende ?
 F996 EB :NB1    wenn nicht
 F998 Ø6 E7 Ø3   Korrektur
*NB1: AF         Rücksprung
```

62

UP Zeilenschaltung

63

```
*ZLS: ØØ E4      Zeilenzähler -1
 F99E EB :ZL1    wenn nicht 0
 F9AØ 48 E3      Anfangswert
 F9A2 BØ 6D      Tastencode := 0
*ZL2: D6 ØC 56   Taste betätigt?
 F9A7 6B :ZL2    wenn nicht
*ZL1: 1C 77      Zeilenschaltung
 F9AB 8B :DR1    Ausführung
```

Drucken eines im Arbeitsregister 1 übergebenen ASCII.
Der Einchip-Mikrorechner setzt nach jedem Rücksetzen P35 auf 1, bevor das Betriebssystem gestartet wird. Daher benötigt unser Programm keine Initialisierung, die Schreibmaschine darf ständig angeschlossen sein. Das Programm kann mit einem beliebigen Arbeitsregistersatz genutzt werden. Außer dem Arbeitsregister 1, in dem der Zeichencode zu übergeben ist, arbeitet es noch mit den Arbeitsregistern Ø,8 und 9. Die weiteren Programme bauen darauf auf.
Das Abschreiben des Bildschirms (Unterprogramm HCB, **Abb. 61**) macht alle Systemunterprogramme für die Bilderzeugung auch für das Drucken nutzbar. Der ASCII-Bildwiederholspeicher (%FDØØ bis %FD7F) dient auch als Druckpuffer. Die erste Schleife ab Marke HC1 sucht die letzte genutzte Bildschirmposition. Die zweite Schleife ab Marke HC2 kopiert das Bild bis zu dieser Position auf den Drucker, wobei das Unterprogramm NBA zum Berechnen der nächsten Bildadresse dient. Zum Drucken einer Zeile ist diese also zunächst auf den Bildschirm auszugeben und dann mit dem Unterprogramm HCB auf die Schreibmaschine zu übertragen.
Das Unterprogramm NBA (**Abb. 62**) erhöht den niederen Teil der Bildspeicheradresse (Arbeitsregister 7) und sichert dabei den lückenlosen Anschluß beim Übergang von einer Bildzeile zur nächsten.
Die Zeilenschaltung erfordert das Druckkommando %77 (Schreibmaschinencode). Zum Anpassen an das übliche A4-Format werden die Zeilenschaltungen mitgezählt, um die Ausgabe zum Blattwechsel zu unterbrechen. Diesem Zweck dient das Unterprogramm ZLS (**Abb. 63**). Es nutzt Arbeitsregister 4 als Zeilenzähler und Arbeitsregister 3 als Speicher für die Zahl der Zeilen je Blatt. Vor Ausgabe des Druck-Kommandos wird der Zähler um 1 verringert. Wenn dabei Ø entsteht, erhält er erneut den Startwert. Außerdem erwartet das Programm dann eine Tastenbetätigung als Bestätigung des Papierwechsels, bevor mit dem Druck-Kommando %77 im Arbeitsregister 1 zum Ausgabeprogramm gesprungen wird. Die Marke DR1 eignet sich als Schnittstelle zur Übergabe von Schreibmaschinencodes (statt ASCII bei DRU).

Wie werden Programmlisten erzeugt?

Zum Drucken von BASIC-Programmen mit BCY (**Abb. 64**) wird das Doppelregister %1Ø genutzt, mit dem das Betriebssystem den BASIC-RAM adressiert. Durch Aufrufen der Systemkomponente BASIC und ggf. Einstellen des Manager-Programms muß das zu druckende BASIC-Programm aktiviert werden. Nach ENTER erhalten %1Ø und %11 dessen Startadresse. Über RESET und PROG können wir nun BCY ab Adresse %F9AD starten. Da BASIC-Zeilen etwa 100 Zeichen enthalten können, setzt es im Querformat eingespanntes Papier und entsprechende Randeinstellungen voraus. Ist einzeiliger Blatttransport gewählt, passen so 40 Zeilen auf einen A4-Bogen. Nach Löschen des Bildschirmes bringt das Programm BCY zuerst die Zeilennumern dann die BASIC-Zeile zur Anzeige und anschließend zum Druck. Der Abbruch erfolgt bei Programmende oder Syntaxfehler mit einem Sprung auf %Ø812.
Für das Drucken wird der Registerpointer auf %4Ø gestellt, um das LIST-Programm (%ØDCC) nicht zu stören. Das Typenrad enthält weder < noch >. Diese Zeichen müssen von Hand nachgetragen werden.
Das Drucken von Maschinenprogrammen (**Abb. 65**) erfordert zuerst die Eingabe von Anfangs- und Endadresse. Das Doppelregister %5E verwaltet die aktuelle Adresse des zu druckenden Programms. Maschinenprogramme haben kurze Zeilen, so daß Hochformat mit 56 Zeilen eingestellt wird. Dabei bleibt auch noch Platz für eine Überschrift. Wie beim Druck von BASIC-Programmen erfolgt ein Halt erst nach Beschreiben eines Blattes. Kommentare können mit der Schreib-

Druck eines BASIC-Programms

```
*BCY: E6 1F 16   BASIC-Pointer
 F9BØ E6 6E ØC   LIST-Adr. H
 F9B3 E6 6F E6   LIST-Adr. L
 F9B6 E6 43 28   Zeilenzahl
 F9B9 E6 44 28   Zeilenzähler
*BC1: D6 Ø8 DD   Bild löschen
 F9BF 31 1Ø      Registerpointer
 F9C1 82 2Ø      Zeilennummer H
 F9C3 42 22      =0 ?
 F9C5 6D Ø8 12   dann fertig
 F9C8 AØ EØ      Adresse +1
 F9CA 82 3Ø      Zeilennummer L
 F9CC 56 E2 7F   Kennbit löschen
 F9CF D6 ØA A3   Anz. Zeilennr.
 F9D2 E6 5B Ø4   Kursor := 4
 F9D5 D6 ØD CC   Anz. Zeile
 F9D8 FD Ø8 12   wenn Fehler
 F9DB AØ EØ      Adresse +1
 F9DD 31 4Ø      Registerpointer
 F9DF D6 :HCB    Drucken
 F9E2 D6 :ZLS    Zeilenschaltung
 F9E5 8B :BC1    nächste Zeile
```

64

Druck eines Maschinenprogramms

```
*MCY: 31 6Ø       Registerpointer
 F9E9 D6 Ø8 DD    Bild löschen
 F9EC D6 23 1F    Eingabe
 F9EF E9 5E       Anf.-Adresse H
 F9F1 F9 5F       Anf.-Adresse L
 F9F3 2Ø 5B       Kursor +1
 F9F5 D6 23 1F    Eingabe
 F9F8 AØ EE       Endadresse +1
 F9FA 31 4Ø       Registerpointer
 F9FC 3C 38       Zeilenzahl
 F9FE 4C 38       Zeilenzähler
*MC1: D6 Ø8 DD    Bild löschen
 FAØ3 D6 26 ØE    Anz. Befehl
 FAØ6 31 4Ø       Registerpointer
 FAØ8 D6 :HC8     Drucken
 FAØB D6 :ZLS     Zeilenschaltung
 FAØE Ø8 5E       Adresse H
 FA1Ø 18 5F       Adresse L
 FA12 24 6F E1    - Endadresse L
 FA15 34 6E EØ    - Endadresse H
 FA18 7B :MC1     wenn nicht Ende
 FA1A 8D Ø8 12    Ende
```

65

maschinen-Tastatur nachgetragen werden, bevor der Computer nach Blattwechsel seine Quittung bekommt.
Damit die Marken im Ausdruck erscheinen, muß PROG mit der Markentabellenadresse des zu druckenden Programms gerufen werden. Nach G F9E7 und L beginnt das Druckprogramm MCY dann wie beschrieben. Sollen statt der Marken die reinen Maschinencodes im Ausdruck erscheinen, kann mit der Markentabellenadresse %FBFF gearbeitet werden.

Können wir auch Hex-Listen drucken?

Für das kompakte Auflisten von Speicherinhalten bietet das Betriebssystem kein geeignetes Bildausgabeprogramm wie etwa LIST (%ØDCC) für BASIC. Deshalb ist das Druckprogramm HEX **(Abb. 66)** etwas aufwendiger. Es beginnt wie MCY mit der Eingabe von Anfangs- und Endadresse des auszudruckenden Speicherbereichs. Es werden 16 byte je Zeile und 16 Zeilen je Block angeordnet. Dabei entsteht ein übersichtliches Druckbild, mit 256 byte (1 Record) zu einem Block. Auf einem A4-Blatt im Hochformat finden bequem drei solcher Blöcke Platz, so daß entsprechend 51 (%33) Zeilen je Blatt eingestellt werden. Die Arbeitsregister 14 und 15 zählen die

Druck eines Speicherbereichs

```
*HEX: 31 6Ø       Registerpointer
 FA1F D6 Ø8 DD    Bild löschen
 FA22 D6 23 1F    Eingabe
 FA25 E9 4A       Anf.-Adresse H
 FA27 F9 4B       Anf.-Adresse L
 FA29 2Ø 5B       Kursor +1
 FA2B D6 23 1F    Eingabe
 FA2E E9 4C       Endadresse H
 FA3Ø E9 4D       Endadresse L
 FA32 31 4Ø       Registerpointer
 FA34 3C 33       Zeilenzahl
 FA36 4C 33       Zeilenzähler
*HE4: EC 1Ø       Blockzeilenzahl
*HE3: FC 1Ø       Spaltenzahl
 FA3C D6 :ZLS     Zeilenschaltung
 FA3F 28 EA       Adresse H
 FA41 D6 :HE1     Hexdruck
 FA44 28 EB       Adresse L
 FA46 D6 :HE1     Hexdruck
 FA49 1C 3A       :
 FA4B D6 :DRU     ASCII-Druck
*HE2: 1C 2Ø       Leerzeichen
 FA5Ø D6 :DRU     ASCII-Druck
 FA53 82 2A       Speicherinhalt
 FA55 D6 :HE1     Hexdruck
 FA58 AØ EA       Adresse +1
 FA5A FA :HE2     16 Spalten
 FA5C 6B EC       Endadresse H
 FA5E 7B ED       Endadresse L
 FA6Ø 22 7B       - Adresse L
 FA62 32 6A       - Adresse H
 FA64 7D Ø8 12    wenn Ende
 FA67 EA :HE3     16 Zeilen
 FA69 D6 :ZLS     Zeilenschaltung
 FA6C 8B :HE4     neues Record
*HE1: D6 :HE5     Hexdruck
*HE5: FØ E2       Halbbytetausch
 FA73 18 E2       ASCII-Register
 FA75 56 E1 ØF    unteres Halbbyte
 FA78 A6 E1 ØA    Ziffer ?
 FA7B 7B :HE6     wenn ja
 FA7D Ø6 E1 Ø7    Korrektur
*HE6: Ø6 E1 3Ø    ASCII
 FA83 8D :DRU     Druck
```

66

Zeilen und Spalten innerhalb des Blocks. Jede Zeile beginnt mit der Adreßangabe, der ein Doppelpunkt folgt. Zum Trennen der Bytes in der Zeile werden Leerzeichen verwendet. Am Zeilenende erfolgt, wenn die Endadresse erreicht ist, der Sprung zum Betriebssystem (Adresse %Ø812), sonst ab Marke HE3 eine Zeilenschaltung als Beginn der nächsten Zeile. Dem Blockende folgt als Trennung eine zusätzliche Zeilenschaltung.
Das hexadezimale Drucken eines Bytes aus Arbeitsregister 2 beginnt mit der Marke HE1. Zur Ausgabe des ersten Zeichens ruft es sich zunächst selbst (D6 :HE5). Mit Halbbytetausch und Löschen der oberen vier Bit werden die darzustellenden vier Bit isoliert und mit Additionsbefehlen in das entsprechende ASCII umgerechnet. Nach dem Druck des ersten Zeichens setzt das Programm mit der Marke HE5 fort. Hier beginnt mit erneutem Halbbytetausch der Druck der zweiten Hexadezimalziffer. Danach kommt der Rücksprung zum Programmpunkt, von dem aus HE1 gerufen wurde.
Vor dem Starten eines Druckprogramms ist nach dem Eingeben mit dem PROG-Kommando T erst zu binden, damit die korrekten Adressen in den Maschinenprogrammen erscheinen. Zum Speichern auf Kassette muß man bei SAVE den Adreßbereich von %F9ØØ bis %FBFF vereinbaren. Er schließt die Markentabelle ein. Werden die bereits gebundenen Druckprogramme wieder von Kassette geladen, sind sie sofort (ohne zu binden) einsatzbereit.

11. Zusätzliche Erweiterungen

Zunächst empfiehlt sich natürlich die Beschäftigung mit den folgenden Beispielen von Anwenderprogrammen (vgl. Teil III). Außerdem geht es mit dem Computerklub in der Zeitschrift JUGEND+TECHNIK weiter: Bezüglich der Entwicklungssystem-Peripherie betrifft das Anwenderprogramme zum Nutzen des Programmierzusatzes für höher integrierte EPROM (bis U 27128) und der Schreibmaschine für graphische Ausgaben. Auch das Implementieren der Programmiersprache FORTH ist für die Veröffentlichung vorgesehen.
Auf dem Gebiet der Gerätetechnik geht es demnächst um das Realisieren von Standard-Schnittstellen (V. 24 und IFSS) mit unserem Computer. Später ist eine Hardware für ein höher aufgelöstes Fernsehbild geplant. Auch weitere Anwenderprogramme werden veröffentlicht. Wer selbst interessante Lösungen findet, sollte sie dem JUGEND+TECHNIK-Computerklub anbieten, damit alle davon profitieren können.

TEIL III: ANWENDERSOFTWARE

Zum Kennenlernen des Computers eignen sich am besten einfache BASIC-Programme. Bei der Eingabe und beim Ausprobieren des „Einmaleins" können wir uns an die Bedienung und die Bildschirmanzeigen gewöhnen. Das Programmbeispiel verbindet das auch noch mit einem Unterhaltungseffekt. Wenn wir später beginnen, selbst Programme zu entwerfen, sind Kenntnisse über das Nutzen des Betriebssystems hilfreich. Zu diesem Zweck folgen tiefergehende Beschreibungen, die wir zunächst aber einfach überblättern können.

12. Nutzbare Unterprogramme des Betriebssystems

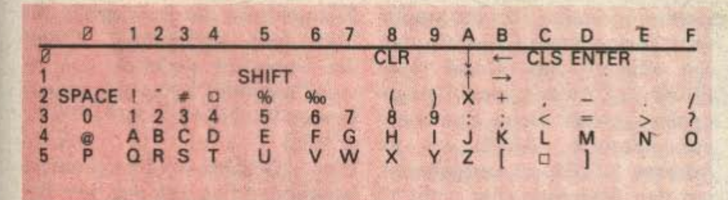
Welche Betriebsprogramme können wir nutzen?

Das Betriebssystem realisiert zur Kommunikation mit dem Bediener verschiedene Anzeigen und Tasteneingaben. Die dazu enthaltenen Unterprogramme lassen sich über CALL-Befehle oder -Anweisungen auch in Anwenderprogrammen einsetzen. Systemprogramme, die RP= %6Ø voraussetzen oder in %12 und %13 Daten übernehmen, sind jedoch BASIC-Programmen nicht zugänglich. **Abb. 67** enthält eine Über-

Abb. 67: Liste allgemein nutzbarer Unterprogramme des Betriebssystems (4 Kbyte)

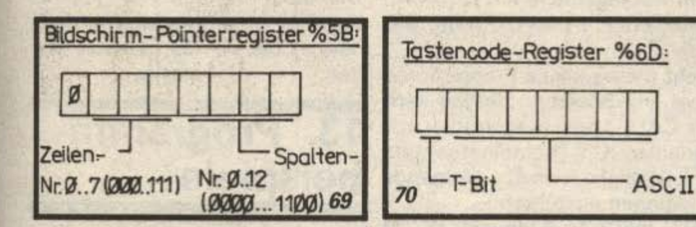
Adresse	Funktion	Bedingung
%Ø824	Tastenabfrage statisch mit Ausführung auf Bildschirm, Kursor (%5B) wird aktualisiert, %5A := ASCII	
%Ø827	Darstellen eines ASCII aus %5A auf dem Bildschirm, Kursor (%5B) wird aktualisiert	
%Ø872	Darstellen eines ASCII aus %5A auf dem Bildschirm, Kursor (%5B) wird aktualisiert, Steuerzeichen werden nicht erkannt	
%Ø875	Erhöhen des Kursors (%5B) um 1 mit ggf. Bildrollen	
%Ø878	Erhöhen des Kursors (%5B) um den Inhalt von %5C mit ggf. Bildrollen	
%Ø8DD	Bild löschen, Kursor (%5B) := Ø	
%ØAA3	Dezimalausgabe aus %12 und %13 ohne führende Nullen	RP = %1Ø
%ØACE	Zeilenschaltung	
%ØAD4	Zeilenschaltung mit Freizeilensperre	
%ØB95	Darstellen eines ASCII aus %5A auf dem Bildschirm, Kursor bleibt unverändert, Steuerzeichen werden nicht erkannt	
%ØC1D	Tastenabfrage statisch ohne Ausführung auf dem Bildschirm %5A := ASCII	
%ØC56	Tastenabfrage dynamisch mittels Tastencoderegister (%6D) : T-Bit und ASCII	
%ØDCC	Anzeige einer BASIC-Zeile ohne Zeilennummer ab Adresse aus %1Ø und %11, die Adresse wird aktualisiert	RP = %1Ø %6E = %ØC %6F =%E6
%ØE92	Dezimalanzeige aus %12 und %13 ohne führende Nullen mit Zeilenschaltung	RP = %1Ø
%2ØB6	Hexadezimalanzeige eines Bytes aus %5D, Kursor (%5B) wird aktualisiert	
%2ØCF	Hexadezimalanzeige der Speicherzelle mit der Adresse aus %5E und %5F, Adresse wird um 1 erhöht, der Kurosr (%5B) wird aktualisiert	
%2ØDB	Hexadezimalanzeige eines Doppelbytes aus %5E und %5F, Kursor (%5B) wird aktualisiert	
%2ØE6	Berechnen der Byteanzahl in %6B eines Befehls mit dem Operationscode aus %5D	
%22E5	Hexadezimaleingabe eines Bytes in %6C mit Tastatur und Anzeige	RP = %6Ø
%231F	Hexadezimaleingabe eines Doppelbytes in %6E und %6F mit Tastatur und Anzeige	RP = %6Ø
%2352	RAM- und Registeranzeige	
%23E6	Programm-Modus	
%26ØE	Anzeige einer Maschinenprogrammzeile mit der Adresse aus %5E und %5F, Adresse und Kursor (%5B) werden aktualisiert, RP := %6Ø	
%268F	Magnetbandausgabe	RP = %6Ø
%27D1	Magnetbandeingabe	RP = %6Ø

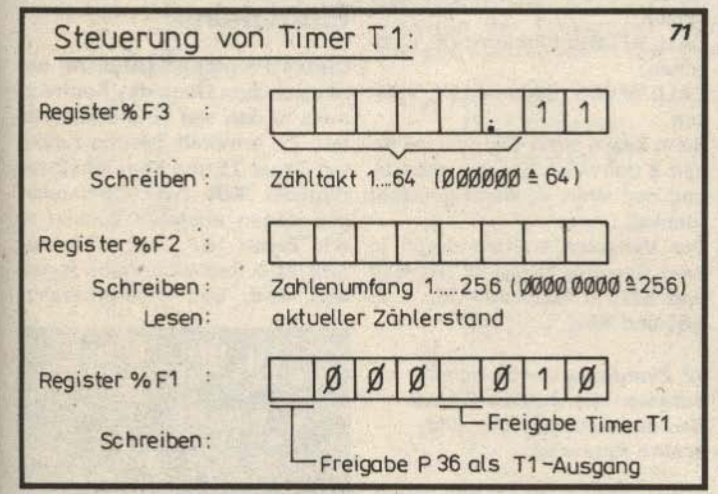
67



	Ø	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Ø									CLR		↓	←	CLS	ENTER		
1						SHIFT					↑	→				
2	SPACE	!	"	#	□	%	‰		(	)	X	+	,	–	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	□	]		

68 Tabelle der Zeichen und Steuercodes (ASCII)





sicht der Startadressen, Funktionen und Bedingungen. Beispiele für das Nutzen dieser Programme geben die Abb. 47, 48, 64, 65 und 66.

Zur internen Zeichendarstellung wird der ASCII zugrunde gelegt. Die **Abb. 68** enthält alle verwendeten Zeichen- und Steuercodes. Der Code %24 steht sowohl für □ als auch für $. Zur Übergabe von ASCII wird das Register %5A genutzt. Beim Darstellen auf dem Bildschirm bestimmt ein Pointer-Register **(Abb. 69)** die Ausgabeposition. Es enthält die Zeile (Ø bis 7) und die Spalte (Ø bis 12) der Bildposition, die mit dem flimmernden Rechteck (Kursor) gekennzeichnet ist und auf die die nächste Zeichenausgabe erfolgt. Der Inhalt dieses Registers ist außerdem gleich dem niederen Adreßteil des ASCII-Bildwiederholspeichers (%FDØØ bis %FD7F), von dem der aktuelle Bildinhalt auch in Form von ASCII gelesen werden kann. Die meisten Betriebsprogramme aktualisieren diese Kursorposition bei jeder Ausgabe, so daß eine Bildschirmadreßangabe überflüssig bleibt. Mit dem Verstellen des Kursors können wir aber den Anfang der nächsten Ausgabe beliebig festlegen. Das funktioniert auch in BASIC-Programmen.

Viele in Abb. 67 dargestellte Komponenten deuten Ein- und Ausgabedaten nicht als Zeichencodes, sondern als Zahlen. Das Register %5D übergibt 8-bit-Zahlen zur hexadezimalen Anzeige. Die Register %5E und %5F werden als Adreßregister beim Bearbeiten von Speicherinhalten genutzt. Sie können auch der Anzeige von 16-bit-Zahlen mit vier Hex-Ziffern dienen. Bei der hexadezimalen Tasteneingabe von Bytes sind %6C, von Doppelbytes %6E und %6F die Übergaberegister.

Für die dynamische Tastenabfrage, die auch ohne Betätigung den Programmablauf nicht blokkiert, müssen wir das Tastencoderegister %6D **(Abb. 70)** auswerten. Es enthält in den unteren sieben Bit den ASCII von der Eingabe bis zur Verarbeitung und als D7 das T-Bit, das den Betätigungszustand der Tastatur mit 1-Pegel, den Ruhezustand mit Ø-Pegel kennzeichnet. Mit dem Löschen der Bits D6 bis DØ quittiert das verarbeitende Programm die Übernahme des letzten Zeichens und gibt damit eine erneute Eingabe frei. Löschen wir auch D7 (T-Bit), gestatten wir das wiederholte Auswerten einer Tastenbetätigung.

Wie werden Töne erzeugt?

Der Einchip-Mikrorechner enthält zwei Timer (TØ und T1), die unabhängig vom Programmablauf durch Frequenzteilung des Quarztaktes (XTAL) mit Dualzählern regelmäßige Impulsfolgen erzeugen können. Das Betriebssystem nutzt TØ zur Bilderzeugung, T1 steht dem Nutzer frei **(Abb. 71)**.

Über P36, wo die Hörkapsel angeschlossen ist, kann mit T1 eine Periodendauer zwischen 2 µs und 32 768 µs (30,5 Hz bis 500 000 Hz) erzeugt werden. Drei Spezialregister bestimmen die Funktion von T1.

Das Register %F3 legt die Periodendauer des Zahltaktes fest. Sie kann zwischen 1 µs und 64 µs als Dualzahl auf den Bitpositionen D7 bis D2 dieses Registers eingestellt werden.

Das Register %F2 bestimmt den Zählumfang. Er kann zwischen 1

und 256 als Dualzahl gewählt werden. Die Periodendauer am Ausgang P36 errechnet sich aus Zähltakt × Zählumfang × 2. So können wir mit dem Laden entsprechender Zahlen in diese beiden Register die Tonfrequenz in weitem Rahmen sehr genau einstellen. Die für einen bestimmten Ton nötigen Zahlen ermitteln wir, indem wir dessen Periodendauer berechnen (reziprok zur Frequenz) und als Produkt zweier Zahlen aus dem Bereich 1 bis 64 bzw. 1 bis 256 und der Zahl 2 in Mikrosekunden angeben.
Mit dem Register %F1 können wir die Tonerzeugung ein- und ausschalten. Das Einschalten gelingt über das Laden der Zahl %8A, das Ausschalten mit 1Ø (hexadezimal %ØA), 2 oder %82.
Beim Lesen der Registeradresse %F2 erhalten wir den aktuellen Zählerstand. Nach dem Laden des Registers %F1 mit %8A (mit Tonausgabe) oder mit 1Ø (ohne Tonausgabe) arbeitet der Timer T1 unabhängig vom Programmablauf, so daß sich sein Inhalt als Pseudo-Zufallszahl eignet. Auch der Inhalt des Registers %54 (zwischen 1 und 12Ø) ist zufällig, wenn das betreffende Programm nicht über eine statische Tastenabfrage (wie INPUT oder GTC) mit der Bilderzeugung synchronisiert wurde. Das gibt die Möglichkeit, die im TINY-MP-BASIC fehlende Funktion RND (Zufallszahlengenerator) zu ersetzen.

Wie erfolgen graphische Bildausgaben?

Die weitestgehend programmtechnisch gestützte Bildschirmsteuerung unseres Computers gestattet ohne Hardware-Ergänzungen graphische Ausgaben. Der Bildwiederholspeicher (%FEØØ bis %FFFF) enthält in jedem Bit die Helligkeitsinformation (1 ≙ dunkel) eines Bildpunktes **(Abb. 72)**. Die Prozeduren SETEB und SETEW gestatten die graphische Ausgabe mit BASIC-Programmen. Schneller arbeiten jedoch Maschinenprogramme. **Abb. 73** stellt allgemein nutzbare Unterprogramme für die graphische Ausgabe als Hex-Liste dar Sie können über DATA oder PROG (vgl. Anfangsmenü) eingegeben werden. Beim Speichern auf Kassette (SAVE) ist ab Adresse %FC8Ø zu vereinbaren, um den Stapelspeicher nicht zu beeinflussen. Das gelingt aus dem Anfangsmenü mit S (SAVE) FC8Ø FCFF PUNKTGRAFIK ENTER.
Steht dieses kleine Programmsystem im Speicher, lassen sich mit CALL-Anweisungen über die Variablen X, Y (Koordinaten) und Z (Helligkeitswert) folgende Funktionen ausführen:
CALL %FCAØ: Bildpunkt (X, Y) setzen,
CALL %FCBØ: Bildpunkt (X, Y) löschen,
CALL %FCBB: Bildpunkt (X, Y) lesen.
Beim Lesen eines Bildpunktes erhält Z den Wert 1, wenn gesetzt, und den Wert Ø, wenn gelöscht (dunkel).
Die Variablen werden durch je zwei Register realisiert (A: %2Ø und %21, B: %22 und %23, ... Z: %52 und %53).

72 Zuordnung der Speicheradressen des Graphik-Bildwiederholspeichers zu den Bildschirm-Positionen

%FEØØ	%%FEØ1	·····	%FEØ7
%FEØ8	%FEØ9	·····	%FEØF
%FE1Ø			·····
·····			·····
·····			·····
·····			·····
·····			·····
%FFF8	%FFF9	·····	%FFFF

```
FCAØ: 7Ø FD 31 7Ø D6 FC CE 6Ø
FCA8: E2 52 32 92 3Ø 5Ø FD AF
FCBØ: 7Ø FD 31 7Ø D6 FC CE 42
FCB8: 32 8B FØ 7Ø FD 31 7Ø D6
FCCØ: FC CE 8Ø 52 BØ 53 72 23
FCC8: EB E3 2Ø 53 8B DF ØC FF
FCDØ: 1C F8 BØ E2 38 51 56 E3
FCD8: 3F 9Ø E3 9Ø E3 1Ø E3 1Ø
FCEØ: E2 22 13 32 Ø2 2C Ø1 38
FCE8: 4F 56 E3 3F 3E ØØ E3 EØ
FCFØ: E2 76 E3 Ø7 EB F7 9Ø E3
FCF8: FØ E3 Ø2 13 B2 3Ø AF ØØ
```

73 Punktgraphik-Unterprogramme als Hexlisting

Beim Nutzen der Punktgrafik mit Maschinenprogrammen erfolgt die Übergabe daher in den jeweils niederen Bytes der betreffenden Variablen (X: %4F, Y: %51, Z: %53).
Die Graphik-Unterprogramme verwenden Register aus dem Systembereich. Daher können sie wie die Programme in Abb. 69 nicht im Schrittbetrieb (mit S, N oder H bei PROG) getestet werden.

13. Programmbeispiele

Einmaleins

Dieses Programm **(Abb. 74)** eignet sich zum Üben des Kopfrechnens in den vier Grundrechenarten. Es ermittelt pseudo-zufällig mit Timer T1 und dem Inhalt von Register %54 zwei Operanden, aus denen ebenfalls zufällig ab den Zeilen 1ØØ, 2ØØ, 3ØØ oder 4ØØ eine Rechenaufgabe formuliert wird. Das Unterprogramm

```
EIN MAL EINS
BIS ZUR 100

WIEVIEL IST
80+1=81
□ RICHTIG ! □
  NOTE: 1
```

```
1Ø  CALL %8DD;
    PRINT "EIN MAL EINS";
    INPUT "BIS ZUR "D;
    LET E=Ø,F=Ø
15  IF D>199
    THEN PRINT "VIEL ZU HOCH";
    WAIT 3ØØ;
    GOTO 1Ø
2Ø  PROC SETRR[%F2,255];
    PROC SETR[%F1,1Ø]
3Ø  WAIT A$M17+1
4Ø  LET A=GETR[%54]$MD+1;
    LET B=GETR[%F2]
5Ø  LET G=B$M4+1,B=B$MD+1
6Ø  IF A<B
    THEN LET C=A
7Ø  ELSE ;
    LET C=B,B=A
8Ø  PROC SETR[%5B,%3Ø];
    PRINT "WIEVIEL IST"
9Ø  GOTO G*1ØØ
1ØØ LET B=B-C,H=B;
    GOSUB 5ØØ;
    PROC PTC[%2B];
    LET H=C;
    GOSUB 55Ø
11Ø IF A=B+C
    THEN GOTO 15Ø
```

```
120 PRINT "# FALSCH ! #";
    LET J=3,I=10;
    GOSUB 600;
    GOTO 160
150 PRINT "* RICHTIG ! *";
    LET E=E+1,J=30,I=1;
    GOSUB 600
160 LET F=F+1;
    IF F<10
    THEN PROC SETR[%58,%40];
    PRINT " ";
    PRINT " ";
    GOTO 30
170 PRINT "   NOTE:";
    PROC SETR[%58,%69];
    LET H=6-(E/2);
    GOSUB 500
180 LET J=10,I=E+1;
    GOSUB 600;
    GOTO 10
200 LET H=B;
    GOSUB 500;
    PROC PTC[%2D];
    LET H=C;
    GOSUB 550
210 IF A=B-C
    THEN GOTO 150
220 GOTO 120
300 LET C=C$M10$04;
    LET B=B/C,H=B;
    GOSUB 500;
    PROC PTC[%2A];
    LET H=C;
    GOSUB 550
310 IF A=B*C
    THEN GOTO 150
320 GOTO 120
400 LET C=C$M10$04;
    LET B=B/C*C,H=B;
    GOSUB 500;
    PROC PTC[%2F];
    LET H=C;
    GOSUB 550
410 IF B/C=A
    THEN GOTO 150
420 GOTO 120
500 IF H>99
    THEN PROC PTC[%31];
    LET H=H-100;
    GOTO 520
510 IF H/10=0
    THEN GOTO 530
520 PROC PTC[H/10+48]
530 PROC PTC[H$M10+48];
    RETURN
550 GOSUB 500;
    PROC PTC[%3D];
    INPUT A;
    RETURN
600 PROC SETR[%6D,0];
    LET K=J
610 PROC SETR[%F2,K];
    PROC SETR[%F1,%8A];
    LET K=K-1;
    IF K>=0
    THEN GOTO 610
620 LET I=I-1;
    IF I>0
    THEN GOTO 600
630 PROC SETR[%F1,10];
    WAIT GETR[%54];
    RETURN
```

5ØØ realisiert die Dezimalausgabe ohne führende Nullen. Die Anweisung 55Ø dient der Anzeige des zweiten Operanden sowie des Gleichheitszeichens (%3D) und der Eingabe des Ergebnisses.

Das Unterprogramm 6ØØ erzeugt eine Melodie aus den Parametern J (Anfangstonhöhe) und I (Anzahl der Wiederholungen). Die WAIT-Anweisung in Zeile 3Ø und 63Ø verbessern die Eigenschaften der Zufallszahlenermittlung, die anschließend (Zeile 4Ø) folgt.

Monophon

Dieses Programm **(Abb. 75)** dient der Tonerzeugung mit dem Timer T1 (Abb. 71). In Zeile 1Ø wird der Zähltakt eingestellt, die Zeile 15 erwartet eine Tastenbetätigung. Um eine hohe Ausführungsgeschwindigkeit zu erreichen, sperrt die Anweisung 2Ø die Interruptannahme. Nach dem Vereinbaren der Tonausgabe über P36 und Sperren von T1 (25) folgt

```
10  CALL %8DD;
    PRINT "MONOPHON";
    PROC SETR[%F3,%23]
15  PROC SETR[%6D,0];
    CALL %C56;
    IF GETR[%6D]=0
    THEN GOTO 15
20  PROC SETR[%FB,%10]
25  PROC SETR[%F1,%82]
30  CALL %C56;
    LET A=GETR[%6D]$A%7F;
    PROC SETR[%6D,0]
35  IF A<42
    THEN GOTO 25
40  GOTO A
42  LET B=101;
    GOTO 130
44  LET B=120;
    GOTO 130
46  LET B=107;
    GOTO 130
47  LET B=95;
    GOTO 130
66  LET B=160;
    GOTO 130
67  LET B=190;
    GOTO 130
68  LET B=202;
    GOTO 130
71  LET B=170;
    GOTO 130
72  LET B=151;
    GOTO 130
74  LET B=135;
    GOTO 130
76  LET B=114;
    GOTO 130
77  LET B=127;
    GOTO 130
78  LET B=143;
    GOTO 130
83  LET B=227;
    GOTO 130
86  LET B=180;
    GOTO 130
88  LET B=214;
    GOTO 130
89  LET B=240
130 PROC SETR[%F2,B];
    PROC SETR[%F1,%8A];
    GOTO 30
```

die dynamische Tastenabfrage mit dem Ablegen des ASCII in der Variablen A. Die IF-Anweisung bewirkt bei Ruhezustand und bei einem Code kleiner als 42 (%2A) das Sperren des Timers (Ton aus) und eine erneute Tastenabfrage. Sonst verzweigt die Zeile 4Ø entsprechend dem ASCII zu einer der folgenden Anweisungen.

Hier erhält B einen Wert, der anschließend als Zählumfang in T1 geladen wird, bevor die Freigabe dieses Timers (Ton ein) erfolgt. Die Zeilennummern und Werte ordnen der unteren Tastenreihe (Y bis /) die ganzen Töne der C-Dur-Tonleiter zu, den darüber liegenden Tasten die Halbtöne. Wie bei einer Orgel erklingen die Töne, solange die betreffende Taste betätigt ist. Eine Verstärkung der Tonausgabe läßt sich in der Regel über den angeschlossenen Kassettenrekorder erzielen, wenn wir ihn in Stellung Aufnahme betreiben, aber die Pausentaste gedrückt lassen.

Mal-fix

Per Tastatur können wir hier mit den Punktgraphik-Unterprogrammen, die natürlich vor Starten des BASIC-Programms geladen werden müssen, beliebige Bilder erzeugen **(Abb. 76)**. Ein Bild läßt sich auch im RAM abspeichern. Zum Auslagern des Bildes auf Kassette müssen wir jedoch die SAVE-Routine des Anfangsmenüs mit dem Adreßbereich EØØØ bis E3FF nutzen, da der betreffende Speicherbereich bei der automatischen Adreßberechnung unter BASIC keine Berücksichtigung findet. Im Programm erfolgt das Adressieren dieses Bereichs mit der Variablen S, die des graphischen Bildwiederholspeichers mit T.

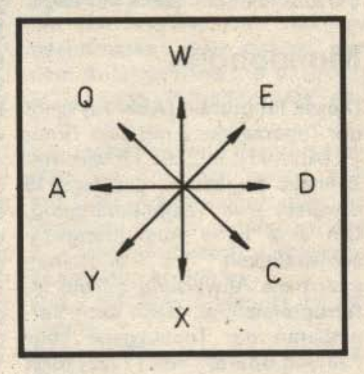
Der Sprungverteiler (Zeile 28) realisiert folgende Tastenfunktionen:

Ø: Bild speichern
K: Kursor bewegen
L: Löschen
M: Malen
K, L und M bestimmen die Be-



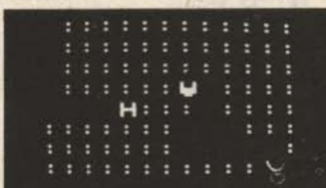

```
10  CALL %8DD;
    PRINT "MAL-FIX";
    PRINT "1=MALEN";
    PRINT "2=HOLEN";
    INPUT "WAHL:"W;
    CALL %8DD
12  LET S=%E228,T=%FE2Ø;
    IF W=2
    THEN LET E=T,F=S;
    GOSUB 3Ø
14  LET X=32,Y=3Ø
16  CALL %C56;
    LET A=GETR[%6D]$A%7F;
    PROC SETR[%6D,Ø]
18  CALL %FCBB;
    IF Z=Ø
    THEN CALL %FCAØ;
    WAIT 1Ø;
    CALL %FCBØ
2Ø  ELSE ;
    CALL %FCBØ;
    WAIT 1Ø;
    CALL %FCAØ
22  IF A<48
    THEN GOTO 16
24  IF C=1
    THEN CALL %FCAØ
26  IF C=2
    THEN CALL %FCBØ
28  GOTO A
3Ø  LET G=236
32  PROC SETEW[E,GETEW[F]];
    LET E=E+2,F=F+2,G=G-1;
    IF G>Ø
    THEN GOTO 32
34  RETURN
36  IF X>63
    THEN LET X=63
38  IF X<Ø
    THEN LET X=Ø
4Ø  IF Y>59
    THEN LET Y=59
42  IF Y<1
    THEN LET Y=1
44  GOTO 16
48  LET E=S,F=T;
    GOSUB 3Ø;
    CALL %8DD;
    END
65  LET X=X-1;
    GOTO 36
67  LET X=X+1,Y=Y-1;
    GOTO 36
68  LET X=X+1;
    GOTO 36
69  LET X=X+1,Y=Y+1;
    GOTO 36
75  LET C=Ø;
    GOTO 16
76  LET C=2;
    GOTO 16
77  LET C=1;
    GOTO 16
81  LET X=X-1,Y=Y+1;
    GOTO 36
87  LET Y=Y+1;
    GOTO 36
88  LET Y=Y-1;
    GOTO 36
89  LET X=X-1,Y=Y-1
127 GOTO 36
```

triebsart bis auf Wiederruf. Die Bewegungen steuern wir mit den Tasten rings um die S-Taste in acht Richtungen **(Abb. 77)**, wobei das Bild entsprechend der Betriebsart bearbeitet wird.



Hase und Wolf

Dieses Programm realisiert ein unterhaltsames Spiel, bei dem der mit einem H symbolisierte Hase ein Feld von Doppelpunkten absammeln muß, ohne sich vom Wolf (W) erwischen zu lassen. Ist ein Feld abgesammelt, geht das Spiel mit einem neuen weiter. Mit jedem Punkt steigt die Bewegungsgeschwindigkeit. Erreicht der Wolf den Hasen, ist das Spiel beendet **(Abb. 78)**.




```
1Ø  LET A=Ø
2Ø  LET B=Ø,D=Ø,E=%7B
3Ø  CALL %8DD;
    PRINT "HASE UND WOLF";
    LET C=1Ø3,F=1Ø1
4Ø  PROC PTC[%3A];
    LET C=C-1;
    IF C>Ø
    THEN GOTO 4Ø
5Ø  PROC SETRR[%F2,%3FF];
    PROC SETR[%F1,1Ø]
6Ø  PROC SETR[%6D,Ø];
    CALL %C56;
    LET G=GETR[%6D]$A%7F
7Ø  LET H=Ø;
    PROC SETR[%6D,Ø]
8Ø  IF G=%5B
    THEN LET H=1
9Ø  IF G=%59
    THEN LET H=-1
1ØØ IF G=%D
    THEN LET H=16
11Ø IF G=%2D
    THEN LET H=-16
12Ø LET J=D;
    GOSUB 5ØØ;
    LET D=J;
    PROC PTC[%48]
13Ø IF D=E
    THEN GOTO 3ØØ
14Ø IF L=%3A
    THEN LET B=B+1,F=F-1;
    PROC SETR[%F1,%8A]
15Ø IF F=Ø
    THEN GOTO 25Ø
16Ø LET K=GETR[%F2];
    IF K=1
    THEN GOTO 21Ø
17Ø IF K=2
    THEN GOTO 2ØØ
18Ø LET H=16;
    IF D<E
    THEN LET H=-16
19Ø GOTO 21Ø
2ØØ LET H=1;
    IF D$A15<E$A15
    THEN LET H=-1
21Ø LET J=E;
    GOSUB 5ØØ;
    LET E=J;
    PROC PTC[%57];
    PROC SETR[%F1,1Ø];
    IF L=%3A
    THEN LET F=F-1
22Ø IF D=E
    THEN GOTO 3ØØ
23Ø IF B<99
    THEN WAIT 99-B
24Ø IF F>Ø
    THEN GOTO 6Ø
25Ø CALL %8DD;
    PRINT "GRATULIERE !";
    PRINT B," PUNKTE"
26Ø PROC SETR[%F3,%23];
    LET T=24Ø
27Ø PROC SETR[%F2,T];
    PROC SETR[%F1,%8A];
    LET T=T*3/4;
    WAIT 2Ø;
    IF T>3Ø
    THEN GOTO 27Ø
28Ø GOTO 3Ø
3ØØ CALL %8DD;
    PRINT "PUNKTZAHL:  "B
31Ø IF A<B
    THEN LET A=B;
    PRINT " ";
    PRINT "** REKORD! ** "
32Ø PRINT " ";
    PRINT "BESTWERT:   "A
33Ø PRINT " ";
    PRINT "COMP JU+TE R";
    PROC SETR[%F3,%23];
    LET T=9
34Ø PROC SETR[%F2,T];
    PROC SETR[%F1,%8A];
    LET T=T*4/3;
    WAIT 3Ø;
    IF T<256
    THEN GOTO 34Ø
35Ø PROC SETR[%F1,1Ø];
    PROC SETR[%6D,Ø];
    LET H=GTC;
    GOTO 2Ø
5ØØ PROC SETR[%5B,J];
    PROC PTC[%2Ø];
    LET L=Ø
51Ø IF J+H$A15>12
    THEN GOTO 55Ø
52Ø IF J+H<Ø
    THEN GOTO 55Ø
53Ø IF J+H>%7B
    THEN GOTO 55Ø
54Ø LET J=J+H,L=GETEB[%FDØØ+J]
55Ø PROC SETR[%5B,J];
    RETURN
```

Die Bewegung des Hasens steuern wir mit der Tastatur (Y: links, X: rechts, –: oben, ENTER: unten). Der Computer führt den Wolf. Dabei dient der Timer T1 als Zufallsgenerator. Während des Spiels quittiert der Computer außerdem jeden erreichten Punkt mit einem Ton, den Bildwechsel und das Spielende jeweils mit einer kleinen Melodie.

Master Mind

Bei diesem Spiel **(Abb. 79)** gilt es, eine vierstellige Zahl, die die Ziffern 1 bis 6 enthalten kann, zu erraten. Eine Ziffer kann dabei auch mehrmals auftauchen. Dennoch muß man in acht Versuchen die richtige Position jeder Ziffer in der vierstelligen Zahl finden. Der Timer T1 wird in den Zeilen 3Ø bis 7Ø genutzt, die zu suchende Zahl zufällig zu ermitteln. Da die Variantenanzahl (1296) den Zählumfang von 256 überschreitet, muß T1 zweimal gelesen werden. Daraus resultieren jedoch keine merklichen Einschränkungen der Zufälligkeit.
Beim Beurteilen des in den Variablen E, F, G und H gespeicherten Tips, durch Vergleichen mit dem Ziel in A, B, C und D, verwendet das Programm zum Teil die Adressierung als Register

```
MASTER MIND
4 MAL 1 AUS 6
 1:2222 *
 2:2333 O
 3:4244 **
 4:4255 ***
 5:4256 **OO
 6:426■
```

```
10  PROC SETRR[%F2,7];
    PROC SETR[%F1,1Ø]
2Ø  CALL %8DD;
    PRINT " MASTER MIND";
    WAIT 1ØØ
3Ø  LET I=GETR[%F2],T=1
4Ø  LET A=I$M6+1,I=I/6
5Ø  LET B=I$M6+1,I=I/6
6Ø  LET C=I$M6+1
7Ø  LET D=GETR[%F2]$M6+1
8Ø  PRINT "4 MAL 1 AUS 6"
9Ø  PROC PTC[32];
    PROC PTC[T+%3Ø];
    PROC PTC[%3A]
1ØØ LET E=GTC$A7,F=GTC$A7;
    LET G=GTC$A7,H=GTC$A7;
    LET I=Ø,S=Ø;
    PROC PTC[32]
11Ø LET J=%21,K=%29
12Ø IF GETR[J]=GETR[K]
    THEN LET I=I+1;
    GOSUB 4ØØ
13Ø LET J=J+2,K=K+2;
    IF J<%28
    THEN GOTO 12Ø
14Ø LET J=%21
15Ø LET K=%29
16Ø IF GETR[J]=GETR[K]
    THEN LET S=S+1;
    GOSUB 4ØØ
17Ø LET K=K+2;
    IF K<%3Ø
    THEN GOTO 16Ø
18Ø LET J=J+2;
    IF J<%28
    THEN GOTO 15Ø
19Ø LET J=I,A=A$A7,B=B$A7;
    LET C=C$A7,D=D$A7
2ØØ IF I=Ø
    THEN GOTO 22Ø
21Ø PROC PTC[%2A];
    LET I=I-1;
    GOTO 2ØØ
22Ø IF S=Ø
    THEN GOTO 24Ø
23Ø PROC PTC[%4F];
    LET S=S-1;
    GOTO 22Ø
24Ø PRINT ;
    LET T=T+1;
    IF J=4
    THEN GOTO 3ØØ
25Ø IF T<9
    THEN GOTO 9Ø
26Ø LET K=%21
27Ø PROC PTC[GETR[K]+%3Ø];
    LET K=K+2;
    IF K<%28
    THEN GOTO 27Ø
28Ø PRINT " DU NASE!";
    PROC SETR[%F3,63]
29Ø GOTO 31Ø
3ØØ PRINT "# RICHTIG ! #";
    PROC SETR[%F3,7]
31Ø PROC SETR[%F1,%8A];
    LET E=64
32Ø LET E=E+4;
    PROC SETR[%F2,E];
    IF E<256
    THEN GOTO 32Ø
33Ø PROC SETR[%F1,1Ø];
    LET E=GTC;
    GOTO 1Ø
4ØØ PROC SETR[%F1,%8A];
    PROC SETR[J,GETR[J]+16];
    PROC SETR[K,8];
    PROC SETR[%F1,1Ø];
    RETURN
```

(Funktion GETR). Das Unterprogramm (4ØØ) erzeugt bei jeder erkannten Übereinstimmung einen kurzen Ton und hakt die betreffenden Ziffern durch Setzen von Bit D4 bzw. Laden mit der Zahl 8 ab. Auf dem Bild erscheinen ein Sternchen (%2A) für jede völlig richtig getippte und ein O (%4F) für jede richtige, aber auf der falschen Position getippte Ziffer.

Autocross

Bei diesem Spiel **(Abb. 80)** kommt es darauf an, das mit einem I auf dem Bildschirm symbolisierte Auto innerhalb der Bahnbegrenzung (O) zu halten. Für die Bewegung des Bildes nutzt das Programm das durch die Ausgabe in die letzte Zeile (PRINT-

```
00180 METER
GEFAHREN !

** REKORD **
REKORDWEITE:
00180 METER
COMP JU+TE R
```

```
1Ø  LET F=Ø
2Ø  CALL %8DD;
    PRINT "* AUTOCROSS *";
    PRINT " ";
    PRINT " ";
    PRINT "AN DEN START!";
    WAIT 15Ø
3Ø  PRINT " ";
    PRINT " ";
    PRINT " ";
    PRINT "000000  00000";
    LET A=%74,C=%37,E=Ø
4Ø  PROC SETR[%F3,255];
    PROC SETR[%F2,3];
    PROC SETR[%F1,1Ø];
    PROC SETR[%6D,Ø]
5Ø  IF E=2ØØØ
    THEN PROC SETR[%5B,%2Ø];
    PRINT "TOLLE FAHRT !";
    GOTO 12Ø
6Ø  IF E<99Ø
    THEN WAIT 99Ø-E/1Ø
7Ø  LET B=GETR[%F2];
    IF B=1
    THEN GOTO 12Ø
8Ø  IF B=2
    THEN GOTO 11Ø
9Ø  LET A=A-1;
    IF A<%7Ø
    THEN LET A=%71
1ØØ GOTO 12Ø
11Ø LET A=A+1;
    IF A>%77
    THEN LET A=%76
12Ø PROC SETR[%5B,A];
    IF E<2ØØØ
    THEN PRINT "OO  OO"
13Ø ELSE ;
    PRINT "OO OO";
    IF E<3ØØØ
    THEN WAIT 3ØØØ-E/2Ø+1
14Ø CALL %C56;
    LET D=GETR[%6D]$A%7F
15Ø IF D=%59
    THEN LET C=C-1
16Ø IF D=%58
    THEN LET C=C+1
17Ø PROC SETR[%5B,C];
    PROC SETR[%6D,Ø];
    IF GETEB[%FDØØ+C]<>%2Ø
    THEN GOTO 19Ø
18Ø PROC PTC[%49];
    LET E=E+1Ø;
    GOTO 5Ø
19Ø PROC PTC[%58];
    PROC SETR[%5B,Ø]

2ØØ PRINT " UNFALL !"
21Ø PRINT E," METER GEFAHREN!"
22Ø PROC SETR[%F1,%8A];
    LET A=Ø
```

```
230 WAIT 20;
    PROC SETR[%F2,6]
240 PROC PTC[8];
    PROC PTC[%20];
    PROC PTC[%40];
    WAIT 20
250 PROC SETR[%F2,12];
    LET A=A+1;
    IF A<13
    THEN GOTO 230
260 IF E>F
    THEN LET F=E;
    PRINT "** REKORD **";
    PROC SETR[%F2,5];
    WAIT 300
270 PRINT "REKORDWEITE:";
    PRINT F," METER"
280 PRINT "COMP JU+TE R"
290 PROC SETR[%F1,10];
    WAIT 800;
    PROC SETR[%F1,%8A];
    GOTO 20
```

Anweisung in 12Ø bzw. 13Ø) ausgelöste automatische Bildrollen. Bahn- und Autoposition werden über die Variablen A und C durch entsprechendes Voreinstellen des Kursors (Register %5B) für die PRINT- bzw. PTC-Anweisung realisiert.
Das „Auto" steuern wir mit den Tasten X und Y. Der Computer bestimmt den Bahnverlauf mit Hilfe des als Zufallszahlengenerator benutzten Timers T1. Die Kollision mit der Bande symbolisiert das Programm mit einem X (statt I), einem Räumfahrzeug mit Martinshorn und der Auswertung der gezeigten Fahrleistung.

Mondlandung

Wegen der graphischen Anzeige des Raumschiffs setzt das Programm **(Abb. 81)** die Punktgraphik-Unterprogramme im Speicher voraus. Es simuliert die Landung eines Raumschiffs auf dem Mond. Angezeigt werden die aktuelle Geschwindigkeit (V), die Flughöhe (H) und der Tankinhalt (T). Die Parameter lassen sich durch Betätigen der Zifferntasten beeinflussen, die die Triebwerke zum Kompensieren der Mondanziehungskraft steuern. Die Tasten wirken so lange, wie sie betätigt werden. Je höher die Ziffer, um so mehr Treibstoff wird verbraucht. Das Programm wartet jedoch eine Eingabe nicht ab.
Ziel ist, durch dosierten Triebwerkseinsatz mit möglichst geringer Geschwindigkeit die Mondoberfläche zu erreichen. Zur Orientierung erzeugt der Timer T1 eine geschwindigkeitsabhängige Tonhöhe.

```
MONDLANDUNG
V= 00012
H= 00004
T= 00392
KRATERTIEFE:
00008 METER!
KUENSTLER !
                --
```

```
10  CALL %8DD;
    PRINT "MONDLANDUNG"
20  LET X=%36,V=0,H=400,T=500;
    PROC SETRR[%F2,255];
    PROC SETR[%F1,%8A]
30  PROC SETEB[%FFFE,0]
40  IF H>609
    THEN GOTO 70
50  LET A=4,Y=H/10
60  CALL %FCA0;
    LET Y=Y+1,A=A-1;
    IF A>0
    THEN GOTO 60
70  PROC SETR[%5B,%10]
80  PRINT "V="V;
    PRINT "H="H;
    IF T>0
    THEN PRINT "T="T
90  ELSE ;
    PRINT "TANK LEER"
100 CALL %C56;
    LET E=GETR[%6D]$A15*3;
    PROC SETR[%6D,0]
110 IF T<E
    THEN LET E=T
120 LET H=H-V-5+(E/2);
    LET V=V+10-E,T=T-E,A=4;
    LET B=1+ABS[V/3];
    PROC SETR[%F2,256/B]
130 LET Y=Y-1;
    CALL %FCB0;
    LET A=A-1;
    IF A>0
    THEN GOTO 130
140 IF H>1000
    THEN PRINT "RAUMSCHIFF";
    PRINT "VERSCHOLLEN !";
    GOTO 210
150 IF H>0
    THEN GOTO 40
160 LET V=V-H;
    PROC SETR[%F1,10];
    PRINT "KRATERTIEFE:";
    PRINT V/5," METER"
170 IF V>50
    THEN PRINT "WELTRAUM-";
    PRINT "ROWDY !";
    GOTO 210
180 IF V>30
    THEN PRINT "SONNTAGS-";
    PRINT "FLIEGER !";
    GOTO 210
190 IF V>10
    THEN PRINT "PROFI !"
200 ELSE ;
    PRINT "KUENSTLER !"
210 PROC SETEB[%FFFE,V];
    WAIT 800;
    GOTO 10
```

Zahlenraten

Zum Erzeugen großer Zufallszahlen lassen sich der Timer T1 und der Bildzeilenzähler Register %54 gemeinsam nutzen. Nach der WAIT-Anweisung zum Aufheben der Synchronisierung mit der Bilderzeugung ermittelt unser Programm **(Abb. 82)** in der Variablen A die zu ratende Zahl zunächst aus dem Inhalt des Register %54 (höheres Byte) und dann zusätzlich aus dem Zählerstand des Timers T1. Der Divisionsrest ($M) mit dem Grenzwert beschränkt das Ergebnis.
Mit beliebig vielen Tips kann der Bediener nun diese in A gespeicherte Zahl erraten. Wenn er sie getroffen hat, wertet das Programm die Zahl der Versuche

```
* ZAHLEN- *
  RATESPIEL

GRENZWERT:
   100

TIP-NR:  00001
   ? ■
```

```
10  CALL %8DD;
    PRINT "* ZAHLEN- *";
    PRINT " RATESPIEL";
    PROC SETRR[%F2,255];
    PROC SETR[%F1,10]
20  PRINT " ";
    PRINT "GRENZWERT:";
    INPUT "  "G
30  WAIT G$M100+1;
    LET Z=0,A=GETR[%54]*256;
    LET A=A+GETR[%F2]$MG+1
40  LET Z=Z+1;
    PRINT " ";
    PRINT "TIP-NR."Z;
    INPUT "  ? "D
50  IF D>A
    THEN PRINT "ZU GROSS !"
60  IF D<A
    THEN PRINT "ZU KLEIN !"
70  IF D<>A
    THEN PRINT " ";
    GOTO 40
80  CALL %8DD;
    PRINT "* RICHTIG ! *";
    PRINT "DIE ZAHL";
    PRINT "HEISST"A,"."
90  PROC SETR[%6D,0];
    WAIT 400;
    PRINT " ";
    PRINT "ANZAHL DER";
    PRINT "VERSUCHE:";
    PRINT Z
100 PRINT " ";
    LET B=0
110 LET B=B+1,G=G/2;
    IF G>0
    THEN GOTO 110
120 IF Z>B
    THEN PRINT "ABER ES GEHT";
    PRINT "NOCH BESSER!"
130 ELSE ;
    PRINT "  REIFE";
    PRINT "LEISTUNG!"
140 PROC SETR[%F2,Z*3];
    PROC SETR[%F1,%8A];
    LET A=GTC;
    GOTO 10
```

aus. Dazu erhält B in der Zeile 11Ø die Anzahl der beim aktuellen Grenzwert nötigen. Hat der Bediener mehr gebraucht, erhält er eine leichte Kritik. Die Zeile 14Ø bewirkt das Erzeugen eines Tones, der um so höher ist, je weniger Tips benötigt wurden.

Pasch

Pasch ist ein Würfelspiel, bei dem in verschiedenen Bewertungskategorien möglichst viele Punkte zu sammeln sind. Unser Programm **(Abb. 83)** bietet das Sammeln von möglichst viel gleichen Zahlen in einem Wurf an. Daher hat es sechs Wertungskategorien, für jede mögliche Ziffer (1 bis 6) eine. Die Ergebnisse speichern die Variablen A bis F.
Gespielt wird mit fünf Würfeln. Der Computer gestattet sechs Versuche mit je drei Würfen. Nach dem ersten und zweiten Wurf jedes Versuchs entscheidet der Spieler, welche Positionen stehen bleiben und welche für den nächsten Wurf verschwinden sollen. Dann ist einzugeben, in welcher Kategorie gewertet wird, welche Zahl gesammelt wurde. Bei einem Pasch (fünf gleiche) zählt der Wurf doppelt und erhält einen Bonus. Ein weiterer Bonus wird gewährt, wenn nach sechs Versuchen mindestens 63 Punkte zu Buche stehen.

```
10  LET L=Ø
20  CALL %8DD;
    PRINT "  PASCH";
    PRINT "1:       2:";
    PRINT "3:       4:";
    PRINT "5:       6:"
30  PROC SETRR[%F2,%657];
    PROC SETR[%F1,1Ø];
    PRINT "WURF   :"
40  LET A=Ø,B=Ø,C=Ø,D=Ø;
    LET E=Ø,F=Ø,I=Ø
50  LET N=A,M=%12;
    GOSUB 2ØØ;
    LET N=B,M=%19;
    GOSUB 2ØØ
60  LET N=C,M=%22;
    GOSUB 2ØØ;
    LET N=D,M=%29;
    GOSUB 2ØØ
70  LET N=E,M=%32;
    GOSUB 2ØØ;
    LET N=F,M=%39;
    GOSUB 2ØØ
80  LET H=Ø,O=Ø,P=Ø;
    LET Q=Ø,R=Ø,S=Ø;
    IF I=6
    THEN GOTO 7ØØ
90  LET H=H+1,N=H,M=%44;
    GOSUB 2ØØ
100 IF O=Ø
    THEN GOSUB 3ØØ;
    LET O=K
110 IF P=Ø
    THEN GOSUB 3ØØ;
    LET P=K
120 IF Q=Ø
    THEN GOSUB 3ØØ;
    LET Q=K
130 IF R=Ø
    THEN GOSUB 3ØØ;
    LET R=K
140 IF S=Ø
    THEN GOSUB 3ØØ;
    LET S=K
150 GOTO 4ØØ
200 PROC SETR[%58,M];
    PROC PTC[32];
    IF N/1Ø=Ø
    THEN GOTO 22Ø
210 PROC PTC[N/1Ø+48]
220 PROC PTC[N$M1Ø+48];
    LET M=M+2;
    RETURN
300 WAIT GETR[%54]$M9+1;
    LET K=GETR[%F2];
    RETURN
400 LET N=O,M=%5Ø;
    GOSUB 2ØØ;
    LET N=P;
    GOSUB 2ØØ;
    LET N=Q;
    GOSUB 2ØØ;
    LET N=R;
    GOSUB 2ØØ
410 LET N=S;
    GOSUB 2ØØ;
    PRINT ;
    IF H=3
    THEN GOTO 51Ø
420 PRINT "LOESCHEN:";
    LET J=GTC$A%7F;
    PROC PTC[8]
430 IF J>53
    THEN GOTO 9Ø
440 IF J<49
    THEN GOTO 9Ø
450 IF J=49
    THEN LET O=Ø
460 IF J=5Ø
    THEN LET P=Ø
470 IF J=51
    THEN LET Q=Ø
480 IF J=52
    THEN LET R=Ø
490 IF J=53
    THEN LET S=Ø
500 GOTO 4ØØ
510 PRINT "ZAHL:";
    LET J=GTC$A7
520 LET K=Ø;
    IF J=O
    THEN LET K=J
530 IF J=P
    THEN LET K=K+J
540 IF J=Q
    THEN LET K=K+Q
550 IF J=R
    THEN LET K=K+R
560 IF J=S
    THEN LET K=K+S
570 IF K=5*J
    THEN LET K=K+K+3Ø;
    PROC SETR[%58,9];
    PRINT "#"
580 IF J=1
    THEN LET A=K
590 IF J=2
    THEN LET B=K
600 IF J=3
    THEN LET C=K
610 IF J=4
    THEN LET D=K
620 IF J=5
    THEN LET E=K
630 IF J=6
    THEN LET F=K
640 LET I=I+1;
    GOTO 5Ø
700 LET K=A+B+C+D+E+F;
    LET K=K/63*5Ø+K
710 PRINT K," PUNKTE";
    IF K>L
    THEN LET L=K;
    PRINT "REKORD!"
720 PRINT "SPITZE:"L," ";
    LET J=GTC;
    GOTO 2Ø
```

RAM-Manager

Alle unsere BASIC-Programme sind höchstens 1 Kbyte lang. Bei Erweiterung der RAM-Ausstattung auf insgesamt 8 Kbyte finden auch sieben Programme gleichzeitig Platz. Der Computer „sieht" aber auch nur das erste, da das Registerpaar 6 nach jedem Rücksetzen den Wert %EØØØ als Anfangsadresse des BASIC-RAM erhält. Indem wir dort eine andere Zahl eintragen, können wir jedoch beliebige andere Adressen als RAM-Anfang definieren.

Abb. 84 zeigt ein geeignetes Programm, das als Vorspann des ersten BASIC-Programms konzipiert ist. Der Computer erwartet nach Ausführen von END ein Kommando, das sich auf die betreffende Scheibe bezieht. Ein Programm wie „Monophon" läßt genug Platz und kann daher als erstes Programm geladen sowie mit dem Manager ergänzt werden. Danach laden wir die übrigen Programme in die Scheiben 2 bis 7.

„Mal-fix" muß aber an die tatsächliche Position angepaßt werden, da sonst ein abgelegtes Bild in der Scheibe 1 landet und dort Schaden anrichtet. Die Zeile 12 weist der Variablen S daher die Zahl %E628 (Scheibe 2), %EA28 (Scheibe 3), ... bzw. %FA28 (Scheibe 7) zu, um die Aufteilung des Managers nicht zu stören.

```
1   PRINT "RAM-MANAGER";
    INPUT "PROGRAMM-NR:"A
2   IF A=1
    THEN GOTO 1Ø
3   IF A<1
    THEN GOTO 1
4   IF A>7
    THEN GOTO 1
5   PROC SETR[6,4*A+%DC];
    END
```

Anhang

Register der Abbildungen (außer Fotos)

Weiterführende Literatur

Schiller: Computerwissen für alle, VEB Fachbuchverlang, Leipzig 1988

Bormann/Rentzsch: Einchip-Mikrorechner, Militärverlag der DDR, Berlin 1989

Kieser/Bankel: Einchipmikrorechner, VEB Verlag Technik, Berlin 1986

Claßen/Oefler: Wissensspeicher Mikrorechner-Programmierung, VEB Verlag Technik, Berlin 1986

Technische Beschreibung Einchip-Mikrorechner-Schaltkreise U881/U882, VEB Mikroelektronik „Karl Marx", Erfurt 1985

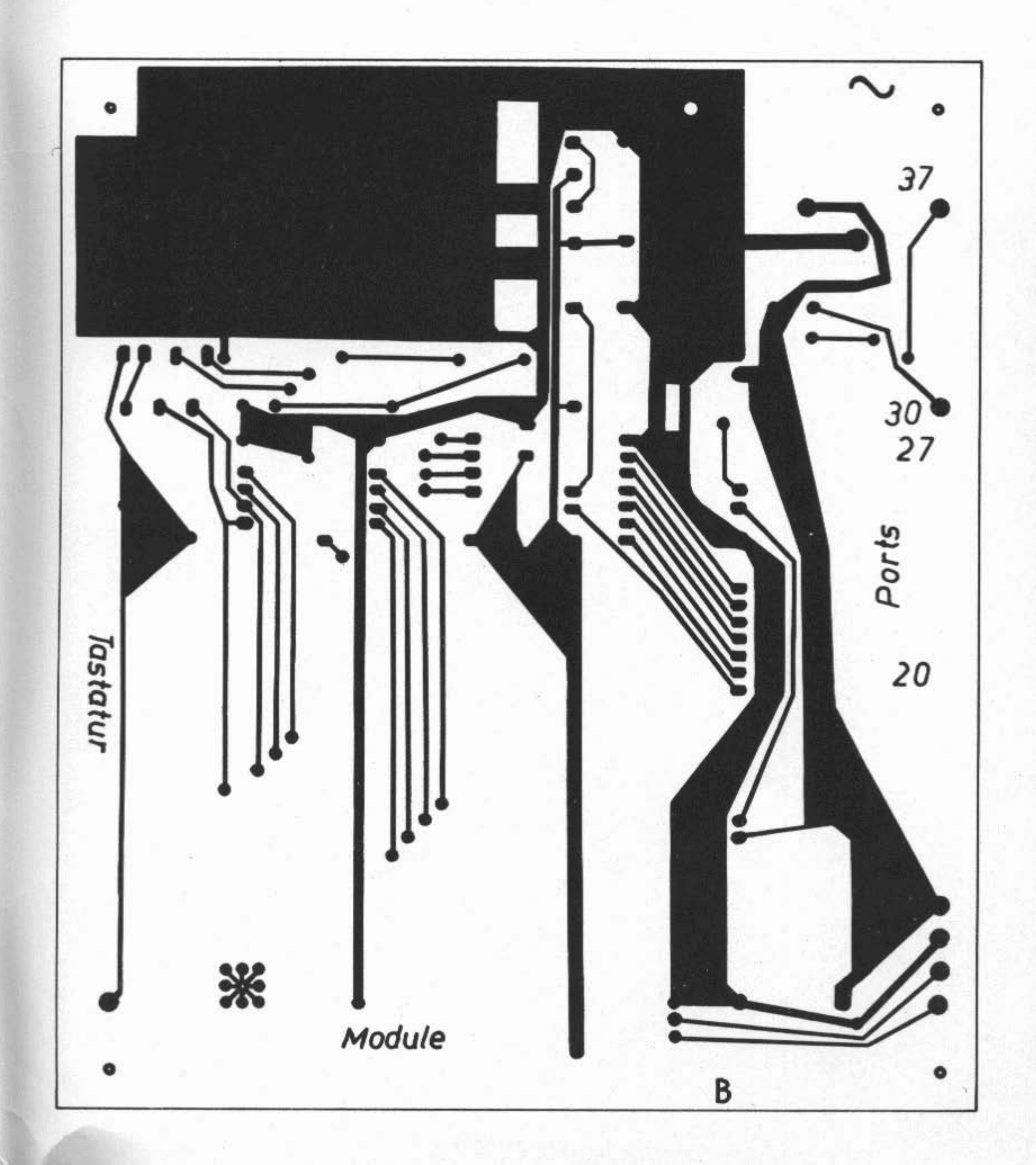
Bennewitz/Podzuweit: Programmierung von Einchipmikrorechnern, Reihe Automatisierungstechnik, Band 215, 2. Auflage, VEB Verlag Technik Berlin 1987

Hoyer: ABC Einchip-Mikrorechner, Zeitschrift JUGEND+TECHNIK, Heft 7/88 bis 6/89

JU+TE-Computerklub, Zeitschrift JUGEND+TECHNIK, Heft 6/87 ff.

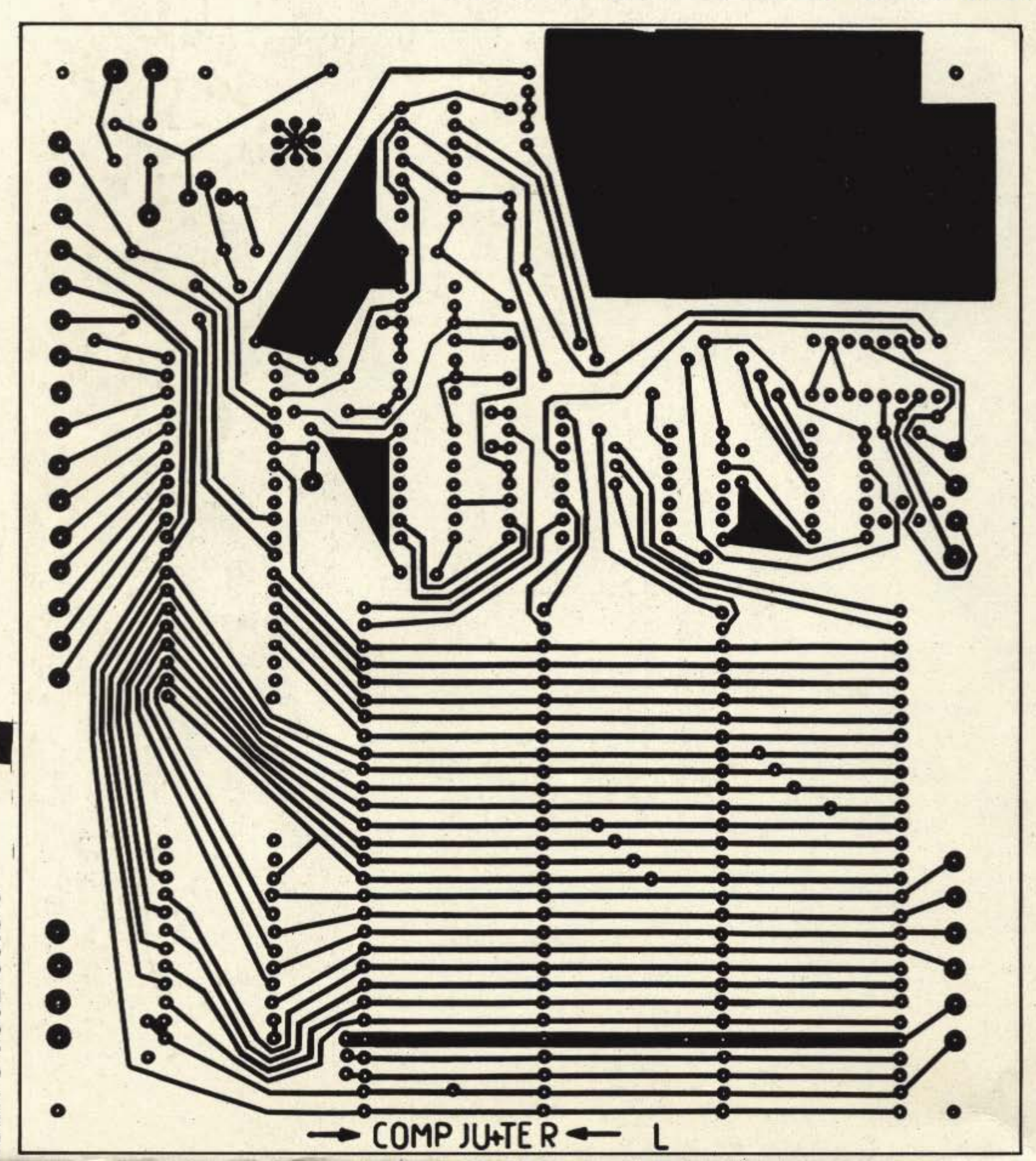
ISBN 3-7302-0613-3

1. Auflage
Druckgenehmigungs-Nr.:
Printed in the German Democratic Republic
Gesamtherstellung:
Druckhaus Schöneweide
Bestellnummer: 684 8955
Titelfoto: Wadim Gratschow
Zeichnungen: Helmut Hoyer, Niels Liebig, Renate Schmidt
Lektor: Norbert Klotz
Gesamtgestaltung: Heinz Jäger
Redaktionsschluß: November 1989
00450



00450

↑ ↓	Trap ! 1	" 2	# 3	¤ 4	% 5	‰ 6	' 7	(8	) 9	Ø	CLR New
=	Q	W Wait	E End	R Return	T Stop	Z	U	I Input	Load O Proc	Save P Print	; + Run
Shift	A	S Gosub	D	F If	G Goto	H Pth	J	[K	L Let	: * Cont	— List
	Y	X	C Call	V	B	N	] M Rem	> ; Then	Else < .	? / Toff	Off Enter



ISBN 3-7302-0613-3